

SCHRIFTLICHE HAUSARBEIT ZUR ABSCHLUSSPRÜFUNG

der erweiternden Studien für Lehrer im

Fach INFORMATIK

**VII. WEITERBILDUNGSKURS IN ZUSAMMENARBEIT MIT DER
FERNUNIVERSITÄT HAGEN**

Eingereicht dem Amt für Lehrerausbildung – Außenstelle Gießen –
Vorsitzender des Prüfungsausschusses: G. Jungermann

**LEGO MINDSTORMS
im WP-UNTERRICHT**



Verfasser: Dr. Wolfgang Seelinger

Gutachter: Dr. Jürgen Poloczek

INHALTSVERZEICHNIS

1.	VORWORT	Seite 3
2.	LERNVORAUSSETZUNGEN	
2.1	Organisatorische und räumliche Voraussetzungen	Seite 4
2.2	Technische Voraussetzungen	Seite 4
2.3	Bild der Kurse	Seite 6
3.	DIDAKTISCHE BEGRÜNDUNG	
3.1	Übergeordnete Ziele des Informatikunterrichts	Seite 7
3.2	Auswahlkriterien	Seite 9
3.3	Objektorientierung als Unterrichtsparadigma	Seite 11
3.4	Übergeordnete Lernziele	Seite 13
4.	DAS LEGO ROBOTICS-INVENTION-SYSTEM 1.5 UND 2.0	
4.1	Die Hardware	Seite 14
4.2	Die Software	Seite 16
5.	METHODISCHE ÜBERLEGUNGEN	Seite 19
6.	DIE UNTERRICHTSREIHE	
6.1	Die erste Doppelstunde	Seite 22
6.2	Die zweite Doppelstunde	Seite 25
6.3	Die dritte Doppelstunde	Seite 29
6.4	Weitere Übungsstunden	Seite 32
6.5	Der Dosenwettstreit	Seite 33
6.6	Möglichkeiten zur Fortsetzung der Unterrichtsreihe	Seite 34
7.	SCHLUSSBEMERKUNGEN	Seite 36
8.	LITERATURVERZEICHNIS	Seite 37
 ANHANG		
	ARBEITSBLATT 1	Seite 1 – 3
	ARBEITSBLATT 2	Seite 1 – 5
	ARBEITSBLATT 3	Seite 1 – 8
	ARBEITSBLATT 4	Seite 1 – 19
	ARBEITSBLATT 5	Seite 1 – 3

1. VORWORT

Die Idee einer Unterrichtsreihe **LEGO-MINDSTORMS** (nicht nur) **im WP-Unterricht** entwickelte sich aus einer Seminarsitzung, in der Herr Dr. Jürgen Poloczek das **Robotics-Invention-System (RIS)** vorstellte. Die Kombination aus Planung und Konstruktion von Robotern sowie deren Programmierung faszinierte alle Seminar teilnehmer. Eine vergleichbare Begeisterung konnte ich im Rahmen der Veranstaltung „Techno (k)now“ am 16. 9. 2002 an der Liebig-Schule in Frankfurt feststellen. Schülerinnen und Schüler der Sekundarstufe II drängten sich am Stand der Firma Kaleidoskop, die einen Parcours für LEGO-MINDSTORMS-Roboter aufgebaut hatte. In kurzer Zeit gelang es sogar Schülerinnen und Schülern, die noch nie am Informatik unterricht teilgenommen hatten, einfache Steuerprogramme zu erstellen und zu testen.

Schnell war von Schülerseite aus der Wunsch nach einer AG formuliert, dem ich gerne nachkam. Die Finanzierung der ersten Baukästen übernahm die humanistische Stiftung, der an dieser Stelle herzlich für ihr Engagement gedankt werden soll. Am 15. 2. 2003, dem Tag der offenen Tür an der Musterschule Frankfurt, konnten bereits die ersten Roboter ihr Können zeigen.

Inzwischen reifte der Entschluss, auch in der Sekundarstufe I eine Unterrichtsreihe mit dem Lego Robotics-Invention-System einzurichten. Da beide WPU-Gruppen, die ich im Schuljahr 2002/2003 unterrichtete, großes Interesse an einer Projektarbeit zu diesem Thema hatten, mussten zunächst organisatorische Hindernisse überwunden werden. Die humanistische Stiftung finanzierte den Kauf von zusätzlichen Baukästen; Herr Poloczek lieh der Musterschule einige Baukästen aus, so dass in den letzten Wochen des Schuljahres für beide Lerngruppen eine ausreichende Anzahl von Baukästen zur Verfügung stand. Ein Übersicht über die verwendeten Ressourcen findet sich in Kapitel 2.2.

Ich werde im Rahmen einer Sachanalyse zunächst die Relevanz einer Gruppen- und Projektarbeit mit **LEGO-MINDSTORMS** Robotern darlegen. Anschließend werde ich, unter Berücksichtigung der Lerngruppen und der allgemeinen Lernziele, einen Vorschlag der methodischen Umsetzung diskutieren. Meine Erfahrung bei der Durchführung der Unterrichtsreihe sind im praktischen Teil der Arbeit zusammengefasst. Er beinhaltet neben einer Analyse auch Vorschläge zur weiteren Vorgehensweise, die ich aus Zeitmangel in diesem Schulhalbjahr nicht mehr habe durchführen können.

Arbeitsblätter und Programmcodes sind im Anhang zusammengestellt. Alle Text- und Bilddateien sowie Literaturangaben als PDF-Files sind auf der beigefügten CD zu finden. Tools zur Änderung von Standard-Konfigurationen befinden sich auf einer Diskette. Die Arbeit dient als Anregung zur Projektarbeit und lädt hoffentlich zur Nachahmung ein.

2. LERNVORAUSSETZUNGEN

2.1 ORGANISATORISCHE UND RÄUMLICHE VORAUSSETZUNGEN

Die Schülerinnen und Schüler der beiden WPU-Kurse (**WahlPflichtUnterricht**) in Jahrgangsstufe 9 und 10 - zu Schuljahresbeginn Klassen übergreifend neu zusammengestellt - unterrichte ich seit Beginn des Schuljahres 2002/2003. In Jahrgangsstufe 9 konnten die Schülerinnen und Schüler zwischen einer 3. Fremdsprachen (Latein, Französisch und Spanisch), Erdkunde und Informatik auswählen. Für Schülerinnen und Schüler der Klassen 10 wurde der Informatikunterricht im WPU-Bereich erstmals in diesem Schuljahr angeboten; d.h. auch diese Schüler haben, wie ihre Mitschüler aus den 9. Klassen, keine Informatikvorkenntnisse aus dem vorherigen Schuljahr.

Der WP-Unterricht wird in beiden Jahrgängen jeweils 2-stündig in Form einer Doppelstunde (5. + 6. Std.) erteilt. Die Teilnehmerzahl ist auf 16 (+1) Schülerinnen und Schüler beschränkt und richtet sich nach der Anzahl der vorhandenen PCs im Computerraum. Diese Einschränkung ist durch traditionelle Lerninhalte im WP-Unterricht gegeben. Sie kann durch die von mir vorgestellte Arbeitsmethode überwunden werden. Bei dieser Unterrichtsreihe erfolgt das Programmieren der Roboter in Gruppen oder arbeitsteilig, so dass nur so viele PCs wie Arbeitsgruppen benötigt werden. Weitauß wichtiger ist das Vorhandensein von zusätzlichen Arbeitsflächen zum Bereitstellen der Materialien, zum Konstruieren und Testen der Roboter.

Verfügt eine Schule über einige Laptops und einen fahrbaren Wagen (Physiksammlung) für die Unterrichtsmaterialien, so kann die Unterrichtsreihe LEGO-MINDSTORMS im WP-Unterricht auch in einem Klassenraum durchgeführt werden.

2.2 TECHNISCHE VORAUSSETZUNGEN

LEGO-MINDSTORMS Roboter können im Handel in Form von RIS Standard-Baukästen erworben werden. Eine andere Möglichkeit ist die Bestellung bei der Firma LPE Technische Medien GmbH¹⁾, die Konstruktions-Sets unter der Bezeichnung „LEGO-MINDSTORMS for Schools“ anbietet. Diese Ausführungen sind etwas besser ausgestattet als die Standard-Baukästen aus dem Einzelhandel. Sie verfügen u.a. über einen zweiten Lichtsensor, eine kleine Lampe, sowie ein Schneckengetriebe. Weiterhin können bei dieser Firma Einzelteile nachgekauft werden.

Die Erfahrungen aus dem Unterricht zeigen, dass viele Schülerinnen und Schüler im Verlauf der Unterrichtsreihe ihre fahrbaren Roboter mit weiteren Fähigkeiten ausstatten wollen (Bsp. Dosenwettstreit). Dafür wird in der Regel ein 3. Motor benötigt, um zusätzliche Greif- oder Kickmechanismen antreiben zu können. Der Motor kann bei der Firma LPE für 19,50 € nachgekauft werden. Zusätzliche Features, wie Rotations- und Temperatursensoren sind ebenfalls erhältlich. Neben den Grundkästen gibt es im Handel zahlreiche Expansions-Sets, mit deren Bauteilen Spezial-Roboter konstruiert werden können.

¹ Firmenanschrift und Internetadresse im Literaturverzeichnis

Zum Lieferumfang der RIS-Baukästen gehört eine Lego-Software. Die LEGO-MINDSTORMS for Schools Baukästen werden ohne Software ausgeliefert. LPE bietet mit der RoboLab-Software eine Alternative zur Standard-Software an.

Die Programmierung der Roboter erfolgt am PC. Nach den Empfehlungen des Herstellers gelten für den Computer folgende Minimalanforderungen:

- Betriebssystem: Windows 98/ME/XP
- Prozessor: Pentium II mit 233 MHz
- RAM: 32 MB
- Grafik: 800 x 600 SVGA
- CD-ROM Laufwerk, Soundkarte, serielle (9-polig) oder USB-Schnittstelle

Die Programme werden mit einem IR-Transmitter vom PC zum Steuergerät (RCX) der Lego-Roboter übertragen. Die älteren Versionen (Version 1.0 und 1.5) werden an eine 9-polige serielle Schnittstelle angeschlossen. Diese IR-Transmitter benötigen eine eigene 9V Batterie. Die neuere Version 2.0 kann sowohl IR-Transmitter mit serieller als auch USB-Schnittstelle ansteuern. USB-Transmitter brauchen keine separate Energieversorgung. Die Reichweite der Sender kann im Menü „EINSTELLUNGEN“ zwischen groß und klein eingestellt werden. Bei der Verwendung mehrerer Sender in einem Raum kann es bei zu großer Reichweite zu Fehlfunktionen aufgrund von Überlagerungen kommen. Es muss dann mit kleiner Reichweite gearbeitet werden.

RCX-Bausteine gibt es in den Versionen 1.0 und 2.0. Die „Firmware 2“, eine notwendige Grundsoftware, konnte auch auf die älteren Versionen ohne Probleme übertragen werden. Die Energieversorgung der RCX-Bausteine erfolgt über 6 Mignonzellen. Da die Motoren einen erheblichen Energieumsatz haben, ist aus Kostengründen die Verwendung von Akkus zu empfehlen. Geeignet sind NiCd- oder Metallhydridakkus ab einer Kapazität von 1000 mAh. Die (zahlreichen) Ladegeräte sollten über eine Entlade- und Erhaltungsfunktion verfügen. Ortsgebundene Roboter können auch direkt über Netzteile versorgt werden, sofern die RCX-Bausteine mit einer Niederspannungsbuchse (9 – 12 V~) versehen sind (LPE-Typen).

Als Grundausstattung für eine Projektarbeit mit Lego-MindStorms Robotern verfügt die Musterschule Frankfurt über:

- 1 Baukasten Robotics-Invention-System 1.5 mit Lego-Standard-Software 1.5
- 5 Baukästen Robotics-Invention-System 2.0 mit Lego-Standard-Software 2.0
- 2 LPE Baukästen Lego MindStorms for Schools mit Lego-Dacta-RoboLab-Software Version 2.0
- 1 Exploration Mars Expansion Set mit Exploration Mars Software
- 3 Ladestationen für jeweils 4x Mignon- und 2x 9V-Akkus
- 100 1,5V Mignon Akkus (Typ AA)

2.3 BILD DER KURSE

In Jahrgangsstufe 9 haben aus 4 Klassen 8 Mädchen und 8 Jungen den Kurs WPU-Informatik gewählt. Die Kursstärke in Jahrgangsstufe 10 beträgt 17 Personen, davon sind 5 Mädchen und 12 Jungen. Einige Schülerinnen und Schüler sind mir aus dem Mathematik- oder Physikunterricht vergangener Jahre bekannt. Ein ausgeprägtes geschlechtsspezifisches Verhaltensmuster, wie es im Anfängerunterricht am PC¹ in den Klassen 5 und 6 noch vorkommt, konnte ich bei diesen älteren Schülerinnen und Schülern nicht mehr beobachten. Lediglich bei der Bildung der Projektgruppen dominierten gleichgeschlechtliche Gruppen. Es bildeten sich nur wenige gemischt geschlechtliche Gruppen, die dann zumeist aus einer Klasse stammten.

Die Interessenlage der Schülerinnen und Schüler variiert sehr stark. Zum einen gibt es eine große Gruppe von aktiven Mädchen und Jungen, denen das Fach Informatik Freude bereitet, zum anderen sind in den Lerngruppen immer auch Schülerinnen und Schüler vertreten, die ihre Erstwahl nicht erhalten haben oder sich mit keinem der Angebote im WPU-Bereich identifizieren konnten. Ziel eines attraktiven Informatikunterrichts sollte es sein, auch die letztgenannten Schülergruppen durch eine entsprechende Themenauswahl in den Unterrichtsablauf einzubinden. Die Erfahrungen zeigen, dass gerade die vorliegende Projektarbeit eher dafür geeignet ist als Unterrichtseinheiten mit Programmiersprachen als Lerninhalt.

In beiden WPU-Kursen erwarben die Schülerinnen und Schüler im Verlauf des Schuljahres Kenntnisse und Fertigkeiten in der imperativen Programmiersprache Turbo-Pascal. Auf eine Einführung der Programmiersprache Delphi wird zu diesem Zeitpunkt bewusst verzichtet, um Neueinsteigern in der Sekundarstufe II den Zugang nicht zusätzlich zu erschweren. Die Schülerinnen und Schüler kennen Aufbau und Struktur von Pascal-Programmen und können, je nach Leistungsstärke, einfache Algorithmen unter Anleitung oder selbstständig entwickeln. Sie kennen Schleifen- und Kontrollstrukturen sowie den Prozedurenbegriff. Aus dem Physikunterricht sind ihnen die elektrischen Bauteile (Motoren, Lampen und Schalter) bekannt. Mechanische Zusammenhänge, wie Übersetzungen mit Riemen und Zahnrädern, sowie Hebelgesetze und Drehmoment gehören an der Musterschule zum Physikcurriculum der Jahrgangsstufen 7 – 9.

¹ Im Rahmen des Deutschunterrichts erwerben Schülerinnen und Schüler an der Musterschule in Jahrgangsstufe 5 und 6 Grundkenntnisse und -fertigkeiten in MS-Word am PC. Der Unterricht wird teilweise mit halbierten Gruppenstärken durchgeführt. Die Einteilung erfolgt häufig geschlechtsspezifisch.

3. DIDAKTISCHE BEGRÜNDUNG

3.1 ÜBERGEORDNETE ZIELE DES INFORMATIKUNTERRICHTS

ANDREAS SCHWILL¹ beantwortet die Frage: „Was ist Informatik?“ mit:

Informatik ist die Wissenschaft von der systematischen Verarbeitung und Speicherung von Informationen, besonders der automatischen Verarbeitung mit Hilfe von Computern.

Diese kurze und prägnante Definition beinhaltet die wortbildenden Bestandteile **Information** und **Automatik**. SCHWILL unterscheidet in der Wissenschaft Informatik sechs Teilgebiete: die theoretische, die praktische, die technische und die angewandte Informatik, ferner Gesellschaftliche Bezüge der Informatik und Didaktik der Informatik. Diese Bereiche sind nicht als disjunkte Mengen zu betrachten, sondern verfügen untereinander über ein vielfältiges Beziehungsgeflecht. Besonders die Didaktik der Informatik greift bei ihrer Stoffauswahl auf die restlichen Teilgebiete zurück, um deren Erkenntnisse für die Aus-, Fort- und Weiterbildung von Kindern und Erwachsenen bereitzustellen. Die Fachdidaktik muss entscheiden, warum und wann welche Inhalte mit welchen Methoden und Zielen in der Schule behandelt werden. Die untere Abbildung² zeigt die Einbettung der Didaktik der Informatik zwischen der Fachwissenschaft Informatik und dem Lernort Schule unter Berücksichtigung von pädagogischen und psychologischen Aspekten.

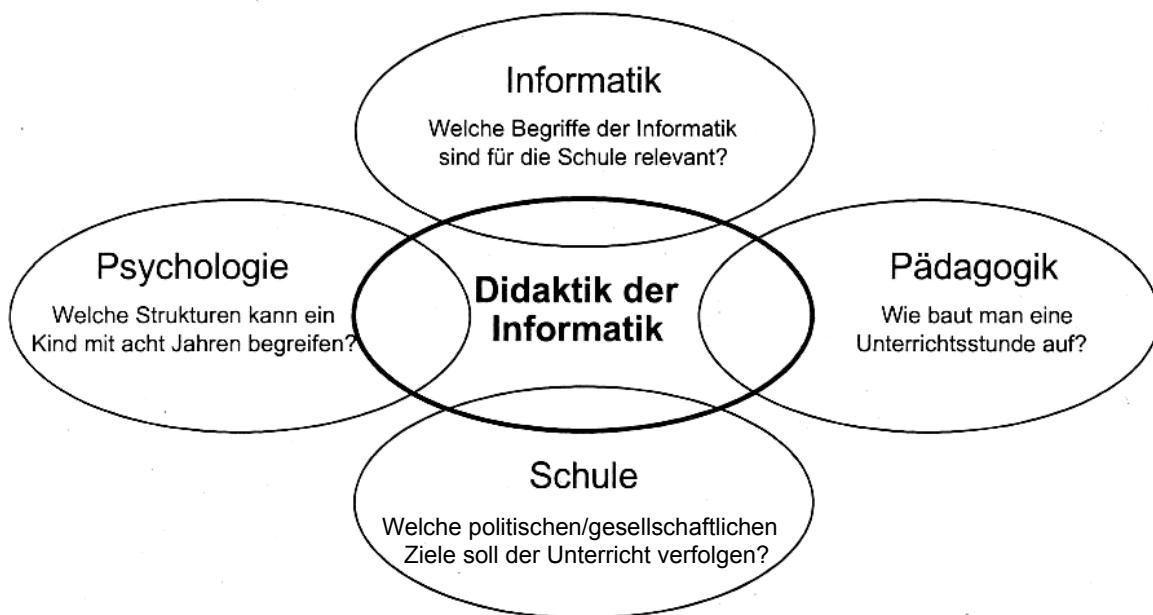


Abbildung: Einbettung der Didaktik der Informatik⁴

Die Auswahlkriterien der Unterrichtsinhalte müssen sich an der zentralen Aufgabe der weiterführenden Schulen, d.h. der fächerübergreifenden Allgemeinbildung, orientieren. BUSSMANN und HEYMANN³ entwickelten einen unter Informatikdidaktikern anerkannten Anforderungskatalog. Danach dient die Bildung:

¹ SCHWILL A. (1996), S. 3

² SCHWILL A. (1996), S. 11

³ BUSSMANN H. und HEYMANN H.-W. (1987), S. 2 – 39

- der Vorbereitung auf zukünftige Lebenssituationen
- der Stiftung kultureller Kohärenz
- dem Aufbau eines zeitgemäßen Weltbildes
- der Anleitung zum kritischen Vernunftgebrauch
- der Entfaltung eines verantwortlichen Umgangs mit erworbenen Kompetenzen
- und der Stärkung des Schüler-Ichs

Die Beiträge des Informatikunterrichts zur Allgemeinbildung werden im Lehrplan Informatik für Gymnasien¹ für die Sekundarstufe II weiter präzisiert:

- Analyse, Beschreibung und Modellierung komplexer Systeme
- Problemlösungsmethoden und ihre Bewertung
- Reflexion des Verhältnisses von Technik und Mensch
- Verantwortungsbewusster Umgang mit Informatiksystemen
- Schöpferisches Denken und Motivation
- Kommunikative und kooperative Arbeitsformen

Infolge des Wandels von einer reinen Industriegesellschaft in eine Informations- und Wissensgesellschaft können die vielschichtigen Strukturen von Wissenschafts-, Wirtschafts- und Gesellschaftssystemen nur noch unter Anwendung informatorischer Methoden und Verfahren verstanden und beherrscht werden. Informatikunterricht hat das Ziel, Schülerinnen und Schülern einen Zugang zu diesen informatorischen Denk- und Arbeitsweisen zu eröffnen und sie damit auf zukünftige Berufs- und Lebenssituationen vorzubereiten. Im Unterricht werden exemplarisch, unter Zuhilfenahme von Hard- und Software, Methoden zur Informationsgewinnung, -strukturierung, -codierung, -verarbeitung, -präsentation und –bewertung vermittelt.

Die Schülerinnen und Schüler erfahren, dass beim Arbeiten mit Informatiksystemen stets Modelle genutzt und Methoden der Modellbildung angewandt werden. Sie lernen die Ergebnisse kritisch zu hinterfragen, da die Modelle die Realität nur stark vereinfacht abbilden. Im Informatikunterricht wird dadurch zum kritischen Vernunftgebrauch angeleitet, der zum Aufbau eines zeitgemäßen, demokratischen Weltbildes beiträgt.

Erst durch die Kenntnis von Anforderungen, Möglichkeiten und Anwendungsfolgen, einschließlich der Risiken und Grenzen des Einsatzes von Informatiksystemen, werden die Schülerinnen und Schüler zum verantwortungsbewussten Umgang befähigt. Sie sammeln Erfahrungen mit den kreativen Gestaltungsmöglichkeiten der Informatikwerkzeuge. Sie erlernen Methoden zur Problemanalyse und –modellierung und erwerben dadurch einen Vorrat an Grundstrategien, die sich auch außerhalb der Schule einsetzen lassen. Der sachbezogene Einblick in das System Computer verhindert das Entstehen von „Computerangst“ und „Computersucht“ und trägt dadurch zur Stärkung des Schüler-Ichs bei.

Die Komplexität der Informatiksysteme erfordert ein arbeitsteiliges Vorgehen im Team. Teamarbeit kann nur dann erfolgreich sein, wenn die Teilnehmer kooperieren und kommunizieren. Nur im Team führt das schöpferische Denken der Einzelnen durch die mehrperspektivische Sicht der Gruppe zum Erfolg. Der Informatikunterricht bietet durch Gruppenunterricht die Möglichkeit, die Teamfähigkeit von Schülerinnen und Schülern zu fördern.

¹ HESSISCHES KULTUSMINISTERIUM (2003), S. 3

3.2 AUSWAHLKRITERIEN

LEHMANN¹ leitet aus diesen allgemeinen Zielen des Informatikunterrichts Kriterien für die Auswahl von Unterrichtsinhalten ab. Er fordert, dass der Informatikunterricht ähnlich wie der Mathematikunterricht auf fundamentalen Ideen aufgebaut sein muss und entwickelt dafür folgende Auswahlrichtlinien:

- *Das Thema muss in verschiedenen Alterstufen vermittelbar sein,*
- *es muss eine längerfristige Relevanz haben,*
- *der Bezug zum Alltag muss gewährleistet sein,*
- *das Thema muss in verschiedenen Bereichen des Fachgebiets vertreten sein.*

Vergleichbare Grundsätze sind auch bei MODROW² zu finden. Er erweitert sie u.a. um nachstehende methodische Aspekte:

- *Die Schülerinnen und Schüler können selbständig in diesem Gebiet arbeiten.*
- *Das Thema eignet sich für arbeitsteiligen Unterricht und/oder Projektarbeit.*
- *Das Thema ermöglicht längere selbstbestimmte Arbeitsphasen.
Der Lehrer muss nicht kurzsichtig den Unterricht führen.*
- *Es gibt nicht nur eine Standardlösung für das Problem.
Die Schüler/innen können das Problem also auf unterschiedlichen Wegen lösen.*
- *Der Zeitbedarf zum Erlernen von Benutzerwissen steht in einem angemessenen Verhältnis zum angestrebten Lernerfolg.*
- *Die Ausgangsproblemstellung kann so einfach gehalten werden, dass sie ohne großes Spezialwissen verstanden und bearbeitet werden kann. Der Zugang zum Thema ist also "naiv" möglich.*

Im Folgenden soll gezeigt werden, dass eine Unterrichtsreihe LEGO-MINDSTORMS diese Kriterien erfüllt.

Der LEGO-MINDSTORMS Baukasten ist mit einer Empfehlung „ab 12 Jahren geeignet“ versehen. Im Rahmen eines Ganztagsangebots werden an der Musterschule im nächsten Schuljahr ab Klasse 7 Kurse mit LEGO-MINDSTORMS Robotern angeboten. Die im praktischen Teil vorgestellte Unterrichtsreihe bezieht sich auf den WP-Unterricht in Jahrgangsstufe 9 und 10 unter Verwendung der RIS 2.0 Software. Für die Sekundarstufe II bietet sich eine Einbindung in höhere Programmiersprachen an. Mittels der ActiveX-Komponente „Spirit.OCX“ können die Roboter z.B. unter den Sprachen Delphi und Visual Basic angesteuert werden. Im Internet findet man weitere Programmierumgebungen wie leJOS und NQC, die einen objektorientierten Ansatz ermöglichen. Ein Zugang ist daher in allen Altersstufen unter technisch-ingenieurwissenschaftlichen, algorithmischen oder objektorientierten Gesichtspunkten möglich, im Sinne des Spiralcurriculums sogar wünschenswert. MODROW³ verlangt,

„dass den Schülerinnen und Schülern die ganze Breite ihrer Möglichkeiten vor Augen zu führen ist, und dazu gehört in der Sekundarstufe II auch ein Überblick über die verschiedenen Wissenschaften. Der riesige Bereich der Ingenieurwissenschaften hat im Fächerkanon des Gymnasiums kein zugeordnetes Fach. Physik - das diese Aufgabe eigentlich übernehmen könnte - wird an Universität und Schule als reines Grundlagenfach betrieben, ist ohne Bezug zu aktueller Technik. Wenn also techni-

¹ LEHMANN E. (1995), S. 29

² MODROW E. in Arbeitsblatt WBK VII, Seminar J. Poloczek (2003) S. 1

³ MODROW E. Virtuelle Lehrerweiterbildung, S. 4

sche Disziplinen mit ihrer anwendungsorientierten Art des Umgangs mit Wissen und ihrer teilweise heuristischen Arbeitsweise von den Heranwachsenden überhaupt als Berufsperspektiven wahrgenommen werden sollen, dann sollte ein Fach sich darum kümmern - und Informatik als einziges technikorientiertes Fach im Gymnasium wäre dafür hervorragend geeignet, weil in seinem Unterricht gerade diese Arbeitsweisen von den Schülerinnen und Schülern erprobt werden können.“

Die Forderung nach einem anwendungsorientierten, themenübergreifenden Informatikunterricht ist ohne Abstriche auch für die Sekundarstufe I gültig. Mit LEGO-MINDSTORMS werden physikalisch-technische Aspekte beim Roboterbau mit mathematisch-algorithmischen Fragestellungen bei der Softwareentwicklung verknüpft. Eine Kombination, die in unserer automatisierten Umwelt alltäglich zu finden ist. Kaum eine neuere technische Errungenschaft kommt heute ohne intelligente Prozessorsteuerung aus. In der Arbeitswelt sind daraus neue Berufsbilder entstanden, wie z.B. das des Mechatronikers oder des Informatik-Ingenieurs.

LEGO-MINDSTORMS bietet durch die Verwendung der klassischen Lego Bausteine eine vertraute Lernumgebung. Gleichwohl wird durch die zusätzlichen mechanischen Komponenten der Bau von beliebig komplexen Maschinen ermöglicht. Die mitgelieferte Software „RIS 2.0“ gewährleistet einen „naiven“ Zugang zur Entwicklung von Steuerungsprogrammen. Das Erlernen einer speziellen Syntax ist nicht notwendig, da das Programmieren über visuelle Symbole mittels der Drag & Drop Methode erfolgt. Das Thema ist deshalb auch für den Anfängerunterricht geeignet. Verfügen die Lerngruppen, wie in Kapitel 2.3 beschrieben, über Kenntnissen und Fertigkeiten im Bereich imperativer Programmiersprachen, so erkennen die Schülerinnen und Schüler bekannte fundamentale Softwarestrukturen wieder und festigen damit ihre Kenntnisse und Fähigkeiten.

Die Unterrichtsreihe Lego-MindStorms ist in eine Einführungs- und eine Projektphase gegliedert. Sie kann um weitere Projekte erweitert werden. In allen Abschnitten können die Schülerinnen und Schüler sehr eigenständig arbeiten und ihr Lerntempo selbst bestimmen. Während der Einführung interagieren die Schülerinnen und Schüler mit dem PC als Trainer, später dient die Programmoberfläche zum Erstellen der Programmcodes. Einfache Roboter, mit Hilfe einer Konstruktionsanleitung zusammengesetzt, sowie elementare Steuerprogramme führen innerhalb kurzer Zeit zu ersten Lernerfolgen. Die Ergebnisse sind überprüfbar (intern & extern) und tragen zur weiteren Motivation bei. Innerhalb eines jeden Lernabschnitts gibt es zahlreiche Möglichkeiten zur Binnendifferenzierung, um auf unterschiedliche Leistungsniveaus innerhalb der Lerngruppen einzugehen.

Aufgabenstellungen, „Challenges“, werden im Verlauf der Unterrichtsreihe immer offener formuliert. Eine der ersten Aufgaben lautet, einen vorgegebenen Robotertyp für eine bestimmten Zeitraum geradeaus fahren zu lassen. Später muss die Aufrichterfordnung, Gegenstände aus einer schwarz umrandeten Fläche zu entfernen, in eine Vielzahl von Schritten zerlegt werden. Der Problemlöseprozess gliedert sich in

- Problem- und Zielformulierung
- Problemanalyse und Modellbildung
- Algorithmierung
- Codierung und Implementierung
- Benutzungsphase

Bei Fehlerfreiheit ist der Prozess abgeschlossen, ansonsten muss er wiederholt werden. Diese Prüfung findet kontinuierlich statt, um Fehler frühzeitig korrigieren zu können. Bei der Modellierung können propädeutisch die Begriffe Klasse, Objekt und Methode aus der Objektorientierung eingeführt und verwendet werden. UML-Zustandsdiagramme helfen bei der Programmerstellung.

Die alte Idee Pestalozzis¹ vom „Lernen mit Kopf, Herz und Hand“ kann in einer Unterrichtseinheit mit Lego-MINDSTORMS Robotern erfolgreich umgesetzt werden. Nach einer Untersuchung der American Audiovisuell Society² speichert ein Mensch 20% des Gehörten, 30% des Gesehenen und 90% dessen, was er selbst getan hat, im Gedächtnis. Dies liegt darin begründet, dass das Gehirn Informationen in Form von Netzwerken speichert. Je vielfältiger die Bezüge sind, um so größer ist die Behaltensfähigkeit. Die wesentlichen Gestaltungsmerkmale der Reihe liegen daher im Bereich der Handlungs- und Problemorientierung, sowie des entdeckenden Lernens. Die Methoden Gruppenarbeit und Projektunterricht unterstützen dieses Vorhaben. Auf sie wird in Kapitel 5 eingegangen.

3.3 OBJEKTOIENTIERUNG ALS UNTERRICHTSPARADIGMA

Das Schulfach Informatik ist, wie die Wissenschaft Informatik, im Vergleich zu den klassischen Naturwissenschaften ein sehr junge Disziplin. Die schulischen Anfänge der Informatik reichen nur etwa 40 Jahre zurück. Trotz dieses kurzen Zeitraums haben bereits mehrere didaktische Paradigmenwechsel³ stattgefunden.

In den 60-er Jahren wurde der Computer oder „Großrechner“ als spektakuläre neue „Maschine“ wahrgenommen. Dies führte zu ersten schulischen Experimenten zur Rechnerkunde. Im Vordergrund stand das Verständnis der Hardware realer EDV-Anlagen und deren Simulation mit Hilfe von logischen Schaltbausteinen. Die Programmierung und Nutzung von Computern war eine esoterische Spezialdisziplin, und wurde nicht als schulrelevantes Thema verstanden.

Dies änderte sich in dem Maße, wie mit zunehmender Miniaturisierung der Systeme immer breitere Kreise von Anwendern mit Computern umgehen, sprich diese programmieren mussten. Einer Empfehlung der Gesellschaft für Informatik folgend, wurde dieser Perspektivwechsel in der Schulpraxis durch die Abkehr vom rechnerorientiertem Ansatz zu einer an Algorithmen orientierten Vorgehensweise nachvollzogen. Dies war nur möglich, weil Mitte der 70er-Jahren die ersten PCs in Form von Apple- oder Commodore-Rechnern auf dem Markt erschienen. Diese Systeme konnten mit der Interpreter-Sprache BASIC programmiert werden. Der Unterricht wurde von engagierten Kollegen erteilt, die sich zumeist autodidaktisch fortbildeten. Deshalb reduzierte sich das Fach Informatik häufig auf einem Programmierkurs, der sich von Schlüsselwort zu Schlüsselwort fortarbeitete. Die Diskussion gesellschaftlicher Auswirkungen der Computertechnik erfolgte zumeist lösgelöst von den behandelten Problemstellungen und wirkten daher aufgesetzt.

¹ Vgl. GUDJONS H. (1987), in TERHART E. (1997), S. 176

² Vgl. GUDJONS H. (1990), IN BASTIAN J. & GUDJONS H. (1990), S. 53

³ Vgl. BAUMANN R. (1996), S. 112

Die 70-er Jahre waren aber auch durch die „Softwarekrise“ geprägt, durch ein unstrukturiertes Vorgehen bei der Softwareentwicklung, das für die wachsende Komplexität der Softwaresysteme nicht angemessen war. Dies führt zur ersten Blütezeit des akademisch fundierten Software-Engineerings. Dementsprechend finden ab 1980 Modellbildungstechniken Eingang in den Informatikunterricht. Das Konzept des anwendungsorientierten Ansatzes hat den Anspruch industrielle Softwareproduktion realitätsnah auf schüleradäquatem Niveau im Unterricht zu imitieren. Algorithmen, die in einer Programmiersprache codiert werden, sind nur noch ein Teilbereich des Softwareentwicklungszykluses. Diese Form von Informatikunterricht ist bis heute aktuell. Sie ist seit 1990 durch einen systemorientierten Ansatz erweitert worden, der noch stärker die Einbettung in Informatiksysteme betont. Im sogenannten Sprachenstreit setzten sich in den 80-er Jahren streng strukturierte Programmiersprachen (imperativ, prozedural) gegenüber BASIC durch.

Seit Anfang der 90-er Jahre setzt sich in der industriellen Software-Produktion zunehmend die Objektorientierung als zentrales Paradigma durch. Zunächst war Objektorientierung primär eine technische Konsequenz der wachsenden Systemkomplexität. Man erkannte, dass große Softwaresysteme am besten mittels einer Zerlegung in lose gekoppelte funktionale Bausteine, die über wohl definierte Schnittstellen miteinander kommunizieren, realisiert werden können¹. Außerdem wurde klar, dass der unkontrollierte Zugriff auf Daten im Hauptspeicher die Ursache der meisten Softwarefehler war. Die Prinzipien der Trennung der Verantwortlichkeiten („separation of concerns“) und Verbergung der internen technischen Details und insbesondere der Datenhaltung („information hiding“) wurden darum zur Leitlinie der Softwareentwicklung im Großen wie im Kleinen²

Die Umsetzung dieses Programms erfordert aber eine ganz andere Denkrichtung als der funktionale Ansatz. Als wichtigster Schritt zu einem erfolgreichen Softwaresystem stellt sich nicht die Wahl des optimalen Algorithmus, sondern die Identifikation der richtigen abstrakten Darstellung des Problems heraus. Die Modellbildung wird zu einem zentralen Thema der angewandten Informatik. Für die schulische Praxis ist diese Entwicklung von besonderer Bedeutung, da sich das Thema „Modelle“ wie ein roter Faden durch den gesamten naturwissenschaftlichen Unterricht zieht. Indem sich die schulische Informatik den Ansatz der Objektorientierung adaptiert, kann sie also einen bedeutenden Beitrag zur Vermittlung von Methodenkompetenz leisten. So glaubt Baumann³, durch objektorientiertes Denken eine Verlagerung der Unterrichtsschwerpunkte von einer Programmierkompetenz zu einer Modellierungs- und Evaluationskompetenz zu erreichen. Hubwieser⁴ schlägt daher vor, bereits im Anfängerunterricht konsequent mit Begriffen aus der Objektorientierung zu arbeiten und entwickelte dafür Unterrichtseinheiten zur Einführung von Grafik- und Textverarbeitungsprogrammen.

In diesem propädeutischen Kontext kann auch die Lego-Software RIS 2.0 eingesetzt werden. Bei der Modellierung des Systems erkennen die Lernenden, dass die einzelnen Komponenten des physikalischen Baukastens für die Zwecke der Programmierung durch ihre Schnittstellen, also die von ihnen angebotenen Signale (Sensoren) und Aktionen (Aktoren) angemessen beschrieben werden können. So

¹ VGL. DENERT E. (1991), KAP. 6 UND 10

² Einen Überblick über die Anwendungen des Objekt-Paradigmas vermittelt BOUZEGHOUB M. ET AL. (1997).

³ BAUMANN R. (1996) S. 281

⁴ HUBWIESER P. (2001) S. 111

könnte Sie können als erster Entwurf („Papiermodell“) eine zentrale Klasse „Mobiler_Roboter“ modelliert werden, die mit ihren assoziierten Klassen „RCX“, „Motor“, „Sensor“ und „Sound“ durch Aufruf von deren zugehörigen Methoden „vorwärts“, „links“, „rechts“, „rückwärts“ und „melde_messwert“ kommuniziert.

Bei der Umsetzung der Aufgabenstellung helfen Zustandsdiagramme, Programmabläufe zu strukturieren und zu dokumentieren. Dadurch wird auch die Kommunikation zwischen den Gruppenteilen bei arbeitsteiliger Vorgehensweise sichergestellt. Weiterhin können nachträgliche Änderungen im Anforderungskatalog als Programmänderungen einfacher integriert werden. Einen Vorschlag zur Umsetzung erläutert Poloczek¹ in seinen Ausführungen zur Steuerung von Lego-MindStorms Robotern mit Delphi.

Die Lego-Software RIS 2.0 lässt eine objektorientierte Programmierung im engeren Sinne nicht zu, da es keine Sprachkonstrukte gibt, die „information hiding“ unterstützen. Das ist für die hardwarenahe Programmierung der Regelfall, da hier der Focus auf der unmittelbaren Ansteuerung von analogen Geräten liegt. In der Tat wird in der Literatur zur Echtzeitprogrammierung als wesentlicher Vorteil der Objektorientierung die Möglichkeit hervorgehoben, Gruppen von imperativen Anweisungen zu bilden, die als abgeschlossene Konstrukte manipuliert werden können, um so die Spezifika der Hardware zu verbergen². In diesem Sinne können die in RIS 2.0 entwickelten Prozeduren im klassischen, imperativen Stil als Methoden der zentralen Klasse „Mobiler_Roboter“ begriffen werden. Dies kann benutzt werden, um das Verständnis des Konzepts „Methode eines Objekts“ zu vertiefen (Kapselung von Funktionalität durch wohl definierte Schnittstelle). Aus dem genannten Grund würde der Einsatz des Java-Derivats leJOS im WP-Unterricht keinen wesentlichen didaktischen Vorteil bringen (dagegen würde das Programmieren erheblich komplizierter werden).

3.4 ÜBERGEORDNETE LERNZIELE

Die Schüler sollen in der Unterrichtsreihe

- die RIS 2.0 Software und Hardware anwenden können
- (imperative) Programmstrukturen wiedererkennen und anwenden können
- (objektorientierte) Problemlösestrategien und
Modellierungsverfahren erlernen und verinnerlichen
- mit den Lego Robotern eine positive Einstellung zur Arbeit am PC erlangen
- fächerübergreifende Zusammenhänge erkennen und nutzen können
- Bezüge der Informatik zur Umwelt erkennen
- Selbstständiges und selbstkritisches Arbeitsverhalten durch Projektarbeit erlernen
- Teamfähigkeit und soziale Kompetenz durch Gruppenarbeit erwerben

¹ POLOCZEK J. (2002)

² Vgl. GREHAM R., MOOTE R. & CYLIAX I. (1998), Kap. 17.

4. DAS LEGO ROBOTICS-INVENTION-SYSTEM 1.5 UND 2.0

Die beiden folgenden Abschnitte führen in verwendete Hard- und Software ein.

4.1 DIE HARDWARE

Ein Lego-MindStorms Baukasten besteht aus über 700 Einzelteilen. Sein Herzstück ist der RCX (Robotics-Command-System), ein programmierbarer Microcomputer. Der RCX verfügt über:

- Hitachi H8 Microcontroller
- mit 16 MHz und 8 Bit CPU
- 16 KByte ROM
- 32 KByte RAM
- internen Zeitgeber
- 10 Bit A/D Wandler
- LC Display
- IR Sender/Empfänger
- 3 Eingänge (Bezeichnung 1, 2, 3)
- 3 Ausgänge (Bezeichnung A, B, C)
- 1 Tonausgabe

Die Lego Firmware unterstützt

- 5 Programmspeicher mit je 10 Tasks und 8 Unterprogrammen
- 32 globalen Variablen
- 16 lokale Variablen je Task
- 4 Systemzeitgeber

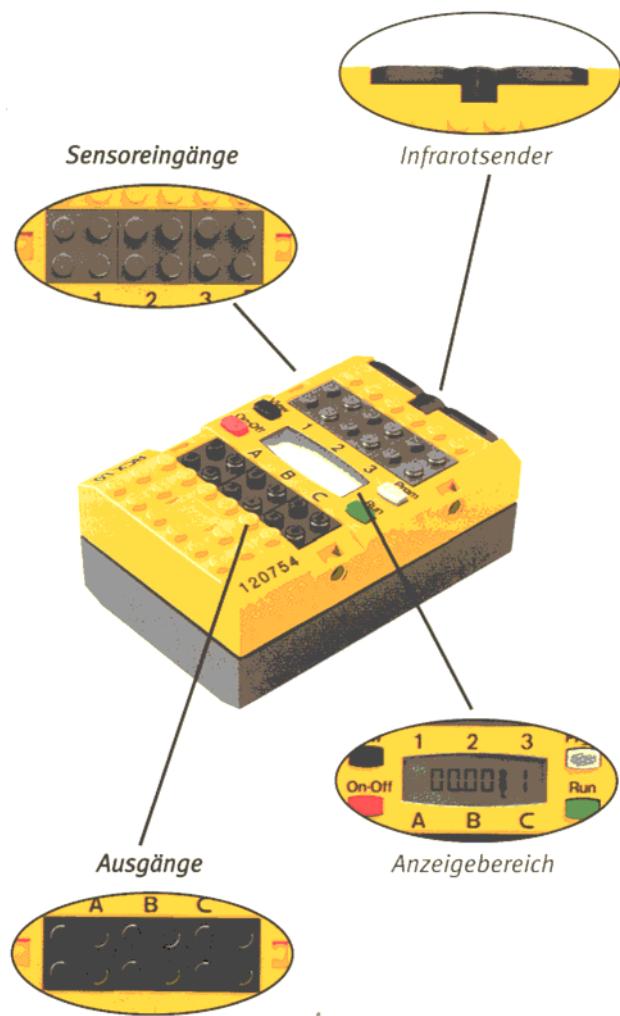


Abbildung: RCX-Baustein¹

Der größte Teil des 32 KByte-Speichers wird von der Firmware und verschiedenen Systemparametern belegt. 6 KByte sind für die eigenen Programme reserviert, so dass für jedes der 5 Programme maximal 1,2 KByte Speicherplatz zur Verfügung stehen. Üblicherweise umfassen Schülerprogramme einige 100 Byte.

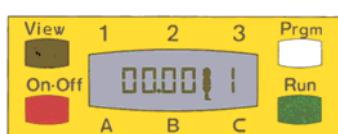


Abbildung: LC-Display

Das Display informiert den Benutzer über den aktuellen Zustand des RCX: Fehlt beim Einschalten die Anzeige der internen Softwareuhr, muss die Firmware neu übertragen werden. Die aktive Programmnummer (1 - 5) wird rechts angezeigt. Läuft die „kleine Figur“, so führt der RCX ein Programm aus. Über die „VIEW“-Taste oder durch Programmbefehle können die Zustände der 3 Eingänge und 3 Ausgänge abgefragt werden.

¹ LEGO MINDSTORMS RIS 1.5, User Guide, S. 7

Abbildung:
9V-Getriebemotor

An den drei Ausgängen können Motoren und Lampen angeschlossen werden. Die 9V-Standardmotoren verfügen über ein eingebautes Untersetzungsgetriebe und drehen maximal 250 U/min. Die Ausgänge können 8 unterschiedliche Leistungsstufen annehmen. Abgeschaltete Motoren können frei laufen oder werden zum Bremsen kurzgeschlossen.

Abbildung:
Lampe

Zur Erforschung seiner Umgebung benötigt ein Roboter Sensoren, um sehen und fühlen zu können. Der RIS-Baukasten ist daher mit Lichtsensoren als primitive Augen und Drucksensoren für den Tastsinn bestückt.

Alle Sensoreingänge werden durch einen 10 Bit A/D-Wandler, d.h. mit max. 1024 Stufen, erfasst. Die Messung der Sensordaten erfolgt alle 100 ms.

Der Drucksensor besteht aus einem kleinen Schalter, der über ein Verbindungskabel mit dem RCX-Baustein verbunden wird. Er wird meistens im logischen Modus verwendet, wobei manchmal auch der Flanken- oder Impulsmodus sinnvoll ist.

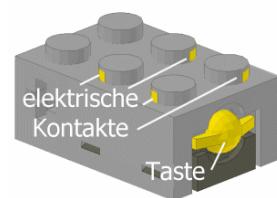


Abbildung: Drucksensor

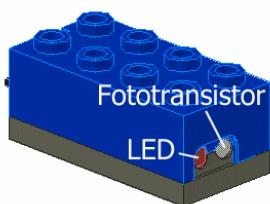


Abbildung: Lichtsensor

Der Lichtsensor besteht aus einer roten LED und einem Fototransistor. Betriebsspannung für LED und Transistor werden vom RCX-Baustein geliefert. Dessen Messbereich liegt zwischen 0,6 und 760 Lux. In den meisten Fällen wird der Lichtsensor im Prozentmodus verwendet, so dass er Werte zwischen 0 und 100 Prozent anzeigt. Da der Lichtsensor seine eigene Lichtquelle enthält, kann er das reflektierte Licht einer nahen Oberfläche messen. Daher können auch stark unterschiedliche Farben detektiert werden.

Der Rotationssensor misst auf opto-elektronischem Weg relative Drehungen. Bei einer Umdrehung der Achse liefert der Sensor 16 Impulse, d.h. ein Impuls entspricht 22,5°. Die absoluten Werte liegen im Bereich von -32767 und +32766. Er kann Umdrehungen bis 500 U/min registrieren. Der Rotationssensor gehört nicht zur Standardausrüstung und sollte unbedingt extra bestellt werden. Er muss im Konfigurationsmenü („HAUPTMENÜ“ → „ERWEITERTE“) initialisiert werden. Der Umdrehungszähler lässt sich beim Programmstart über „KLEINE BLÖCKE“ → „ZURÜCKS.“ → „DREHUNG ZURÜCKS.“ auf den Wert Null zurücksetzen.



Abbildung: Rotationssensor

Es werden noch weitere Spezialsensoren, wie z.B. Temperatursensoren und Geräuschmesser, angeboten. Im Internet finden sich zusätzlich zahlreiche Hinweise zum Eigenbau von Sensoren. Da sie in den Aufgaben dieser Arbeit keine Verwendung finden, soll auf sie nicht eingegangen werden.

4.2 DIE SOFTWARE



Abbildung: HAUPTMENÜ RIS 2.0

Im Lieferumfang des LEGO MINDSTORMS Robotics-Invention-System (RIS) ist eine spezielle Software enthalten. Die geplante Unterrichtsreihe bezieht sich auf die derzeit aktuelle Version 2.0. Auf relevante Unterschiede zwischen der älteren Version 1.5 und der neueren Version wird jeweils hingewiesen. Die unterschiedlichen Versionen sind abwärtskompatibel, d.h. Programme der Version 1.5 (*.rcx) können von Version 2.0 (*.lsc) gelesen und konvertiert werden.

Beide Softwareversionen beinhalten eine interaktive Einführungstour mit Videosequenzen und Aufgabenstellungen. Die Programmierumgebung zur Erstellung der Steuerprogramme arbeitet mit einer visuellen Programmiersprache, d.h. die einzelnen Anweisungen werden durch Symbole in Form von Legosteinen dargestellt und mittels der Drag & Drop Methode zusammengefügt. Dadurch werden die Fehlermöglichkeiten wegen falscher Syntax stark eingeschränkt.

Die RIS-Versionen starten bei einer Neumeldung automatisch mit einer Einführungs-„TOUR“. Gleichzeitig wird für jeden Benutzer bei Version 2.0 unter dem Verzeichnis „...\\LEGO MINDSTORMS\\RIS 2.0\\USERS“ ein Ordner mit einem Benutzernamen angelegt. Eine eigene Konfigurationsdatei „USER“ führt Protokoll über den aktuellen Stand (Trainingsstatus 0..1000) innerhalb der Einführungstour und speichert spezielle Editoreinstellungen ab. Ein Unterordner „VAULT“ dient zur Aufnahme eigener Programme. Die „VAULT“-Verzeichnisse der anderen Benutzer sind nicht geschützt und können alle über die Editor-Funktion „DATEI“ → „ÖFFNEN“ geöffnet und bearbeitet werden. Die Verwaltung und Sicherung von Programmen ist deshalb auch auf zentralen Servern möglich.

Beide RIS-Versionen beinhalten eine Option, die ein Überspringen der Einführungstour ermöglicht (Version 1.5 „INFO“-Option bei gedrückter Strg-Taste wählen; Version 2.0 Menüoption bei Anwahl). Ohne diese Möglichkeit kommt es in PC-Räumen mit Wächterkarten zu Schwierigkeiten, da bei jedem Neustart die Standardordner mit den Benutzernamen neu angelegt werden und daher die Tour immer erneut startet. An der Musterschule ist Version 2.0 über ein virtuelles CD-ROM-Laufwerk installiert und damit netzwerkfähig. Mit der Windows-Taste kann jederzeit auf den Windows-Bildschirm umgeschaltet werden, ohne die Lego-Software schließen zu müssen.

Das Robotics-Invention-System 2.0 verfügt über einige Verbesserungen gegenüber seiner Vorgängerversion 1.5. Im Bereich Einführungs-„LEKTIONEN“ wurde der aktive Bildschirmausschnitt vergrößert und dadurch das gemeinsame Arbeiten mehrerer Schülerinnen und Schüler vor einem Bildschirm ermöglicht. Visuelle und vor allem Audio-Effekte bei den Benutzeroberflächen wurden zu Gunsten von informativeren Auswahlmenüs

reduziert. Die Trainingslektionen sind sehr ausführlich gestaltet. Man merkt ihnen die Herstellerempfehlung „geeignet ab 12 Jahre“ deutlich an. LEGO MINDSTORMS ist daher auch für den Anfängerunterricht in Informatik oder für Projektwochen mit Schülerinnen und Schüler ohne Programmervorkenntnisse geeignet. Einzelne Lektionen können in Version 2.0 beliebig wiederholt oder aber auch, bei entsprechendem Vorwissen, übersprungen werden. Die zum Lieferumfang gehörende Constructopedia beinhaltet für die Grundtypen ausführliche Bauanleitungen, sowie wertvolle Konstruktionstipps für Sonderfunktionen. Im Internet findet man zahlreiche Web-Seiten zu diesem Thema (siehe Anhang Arbeitsblatt 5).

Zu jeder Trainingslektion gibt es Übungsaufgaben. Diese „CHALLENGES“ sind nach Robotertypen („ROVERBOT“, „ACROBOT“ und „INVENTORBOT“) sortiert und jeweils in drei Schwierigkeitsstufen gegliedert. Ziel ist die Bewältigung einer vorgegebenen Problemstellung, für die die Roboter programmiert werden sollen. Die Anforderungen an die Roboter müssen unter Berücksichtigung ihrer konstruktiven Merkmale in eine detaillierte Folge von Programmschritten zerlegt werden. Die anschließende Testphase zeigt, ob die Modellierung erfolgreich und vollständig war. Gegebenenfalls muss der Entwicklungszyklus mehrfach durchlaufen werden. In Kapitel 6 wird die genaue Planung und Vorgehensweise für eine Unterrichtsreihe unter Einbezug der Lego Trainingslektionen und –aufgaben detailliert erläutert.

Zusätzlich werden unter der Menuoption „LEKTIONEN“ → „PRO-AUFGABEN“ weitere 6 Robotertypen („PANZERSCHRANK“, „KÜHLSCHRANK-FRED“, „MALBOT“, „LIEFERBOT“, „SORTIERBOT“ und „ROBOTERARM“) vorgestellt. Die Aufgaben richten sich an erfahrene („Professionals“) Schülerinnen und Schüler. Die vorgestellten Roboter sind mechanisch aufwändiger zu konstruieren und anspruchsvoller zu programmieren. Die Video animierten Anleitungen enthalten hier Aufgabenstellung, Konstruktionshilfen und Kontrollpunkte. Weiterhin gibt es Expansion-Sets, die neben speziellen Lego-Bauteilen und einer eigenen Constructopedia auch eine CD-ROM mit weiteren Aufgaben beinhalten. Auf das „EXPLORATION-MARS-EXPANSION-SET“ wird am Ende der Unterrichtsreihe eingegangen werden. Selbstverständlich können auch eigene Robotertypen im Unterricht ersonnen, entwickelt, konstruiert, programmiert und getestet werden.



Abbildung: Menü „TRAININGSLEKTIONEN“ RIS 2.0

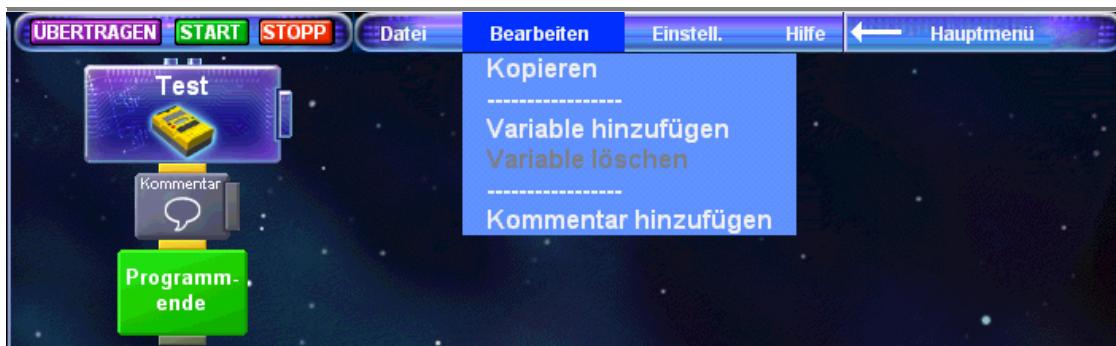


Abbildung: Neue Kopfzeile Programmoberfläche mit „BEARBEITEN“ Menü RIS 2.0

Wesentliche Veränderungen wurden auch im Editor vorgenommen. Die Programmoberfläche ist übersichtlicher geworden. In der Kopfzeile stehen jetzt Drop-Down-Menüs für „DATEI“, „BEARBEITEN“, „EINSTELLUNGEN“ und „HILFE“ zur Verfügung. Die Programme können mit Hilfe von Buttons auf den RCX übertragen, gestartet oder gestoppt werden. Erstmalig ab Version 2.0 können die Programmcodes mit Kommentaren und einem Programmende-Zeichen versehen werden. Eine weitere Art, Programme zu dokumentieren, besteht mit Hilfe des Logbuchs, einem Memofeld, welches beim Speichern bzw. Öffnen erscheint. Damit können jetzt sowohl einzelne Programmschritte, als auch vollständige Programme mit ergänzenden Texten erläutert werden.



Am linken Rand des Editors befinden sich in diversen Untermenüs die verschiedenen Programmelemente. Sie wurden in der Version 2.0 für den inländischen Markt mit deutschen Bezeichnungen versehen. Eine Änderung, die Anfängern entgegenkommt, aber für WPU-Schülerinnen und Schüler der Klassen 9 oder 10 nicht von besonderer Relevanz ist. Wichtiger ist das Vorhandensein einer deutschsprachigen Hilfefunktion, da die Fremdsprachenkenntnisse in der Sekundarstufe I in der Regel zum Verstehen zusammenhängender Fachtexte noch nicht ausreichen.

Die Programmelemente sind in funktionale Gruppen zusammengefasst, die in den unterschiedlichen Trainingslektionen behandelt werden. „GROÙE BLÖCKE“, „KLEINE BLÖCKE“ und „EIGENE BLÖCKE“ beinhalten u.a. Kommandos zur Steuerung der Ausgänge. „GROÙE BLÖCKE“ sind roboterabhängig definiert. Sie entsprechen den Methoden bei objektorientierter Betrachtungsweise. „GROÙE BLÖCKE“ sind aus Kontrollstrukturen und „KLEINEN BLÖCKEN“ zusammengesetzt. Die Anwender können auch selbst „GROÙE BLÖCKE“, sie werden als „EIGENE BLÖCKE“ bezeichnet, zusammenstellen. Durch das Zusammenfassen werden die Programme übersichtlicher. „EIGENE BLÖCKE“ können mehrfach im Programm, allerdings nicht rekursiv, verwendet werden. Aus imperativer Programmiersicht sind sie mit Prozeduren vergleichbar. In den Funktionsgruppen „WARTEN“, „WIEDERHOLEN“, „JA ODER NEIN“ werden Kontrollstrukturen zur Verfügung gestellt. Die ein- oder zweiseitigen Abfragen greifen auf Variable, Timer oder Sensoren zurück und werden sequentiell in den Programmcode eingearbeitet. Verzweigungen werden auch bildlich dargestellt.

Abbildung: Kommando- und Kontrollstrukturen Programmoberfläche RIS 2.0

Eine Besonderheit stellt die letzte Funktionsgruppe „SENSOREN“ dar, die eine Quasi-Interruptsteuerung des Hauptprogramms ermöglicht. Die Bausteine werden seitlich am Hauptprogramm platziert. Bei Erfüllung der logischen Bedingung wird das Hauptprogramm sofort unterbrochen, das entsprechende Unterprogramm durchgeführt und anschließend in das Hauptprogramm zurückgesprungen. Rücksprungstelle ist der Beginn des Programmbefehls, an dem das Hauptprogramm unterbrochen wurde. Diese Art der Ereignissteuerung ist unter dem Blickwinkel eines imperativen Programmierstils unübersichtlich. Mit den zur Verfügung stehenden Kontrollstrukturen können alle Sensorabfragen auch in sequentielle Programmstrukturen implementiert werden.

5. METHODISCHE ÜBERLEGUNGEN

In den Hessischen Lehrplänen¹ der Informatik werden bei den Arbeitsweisen folgende Aspekte hervorgehoben:

Die aktive Auseinandersetzung mit den Inhalten und Methoden der Informatik erfordert eine intensive Nutzung der schulischen Computeranlagen. ... Die Interaktivität des Mediums Computer ermöglicht in besonderer Weise selbständiges und selbsttätiges Lernen der Schülerinnen und Schüler. ... Soziale Kontexte werden durch Gruppen- und Teamarbeit geschaffen. ... Problemorientierung ... lässt sich sinnvoll mit themen- und projektorientierten ... Unterrichtsphasen umsetzen.

SCHWILL² hält den Informatikunterricht besonders für die Projektmethode geeignet, da diese der alltäglichen Arbeitsweise von Informatikern entspricht. Hierbei liegt der Schwerpunkt innerhalb der Schule auf dem Erwerb sozialer Arbeitsmethoden, während die Industrie zuvorderst an einer Leistungsoptimierung interessiert ist. Eines der wichtigsten Lernziele ist daher die Erziehung zu kooperativer Arbeit und die Fähigkeit zur Konfliktlösung. Miteinander Lernen bedeutet auch, die erworbenen Kenntnisse und Fertigkeiten nicht nur für sich selbst einzusetzen, sondern auch anderen in der Gruppe zur Verfügung zu stellen, um das Gruppenprojekt erfolgreich abschließen zu können.

Projektunterricht in seiner reinen Form, d.h. Aufhebung der starren Stundenstruktur zu Gunsten einer realitätsnahen Arbeitsweise, kann nur im Rahmen von schulischen Projektwochen verwirklicht werden. Im Normalfall – Regelunterricht - lassen sich aber projektorientierte Vorhaben durchführen, die wesentliche Momente von Projektunterricht integrieren. BAUMANN³ sieht in Projekten den Kulminationspunkt einer längeren Kurseinheit und möchte die Methode von Anfang an in den Informatikunterricht integriert sehen. Dieser Interpretation von Projektarbeit als größeres Softwarevorhaben, das in Gruppenarbeit durchgeführt wird, schließe ich mich an.

Röhner⁴ beschreibt den zeitlichen Ablauf eines Softwareprojektes (Abbildung nächste Seite) durch ein Phasenmodell, das in der Literatur als „software life cycle“ bezeichnet wird und empfiehlt eine Übertragung auf den Projektunterricht. Das Modell beschreibt den vollständigen Weg einer Software von der Problemstellung bis

¹ HESSISCHES KULTUSMINISTERIUM. (2003), S. 5

² SCHWILL, A. (1996), S. 59

³ BAUMANN, R. (1996), S. 191

⁴ RÖHNER, G. (2002), S. 3

zur Produktabnahme. Ganz im Sinn eines systemorientierten Ansatzes müssen alle Phasen erfolgreich durchlaufen werden. Ansonsten muss der Durchlauf in Teilbereichen wiederholt werden.

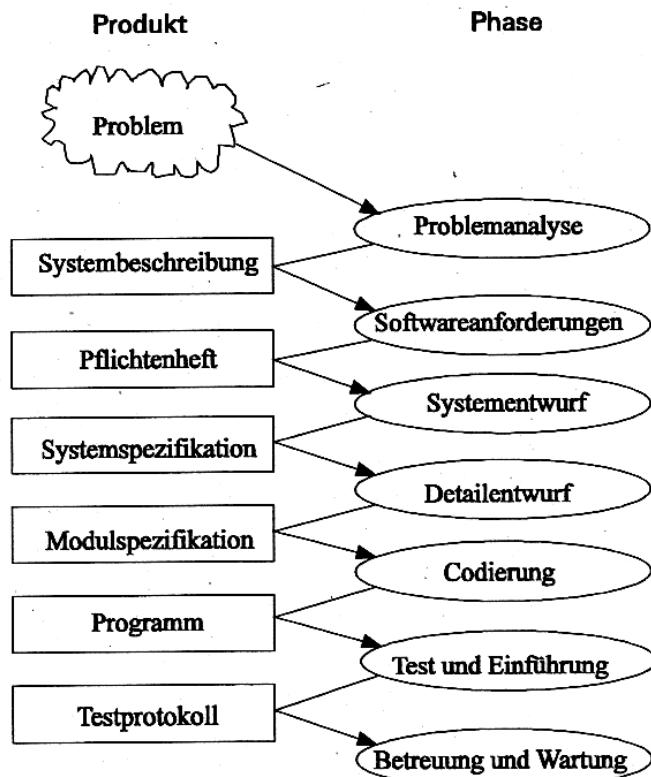


Abbildung: Software life cycle

Bei Gruppenarbeit muss zwischen arbeitsgleicher und arbeitsteiliger Vorgehensweise unterschieden werden. Arbeitsgleich bedeutet, dass alle Gruppenmitglieder gemeinsam an einer Aufgabenstellung arbeiten, Ansätze und Strategien diskutieren, um zu einer gemeinschaftlichen Lösung zu gelangen. Erst dann kann zeitökonomisch arbeitsteilig mit Konstrukteuren (Roboterbau) und Programmierern (Softwareentwicklung) fortgefahrene werden. In regelmäßigen Arbeitssitzungen müssen im Gruppenplenum die Ergebnisse der arbeitsteiligen Vorgehensweise koordiniert und abgestimmt werden.

Gruppenarbeit bedeutet für zurückhaltendere Schülerinnen und Schüler, die das Plenum scheuen, eine Möglichkeit, innerhalb der Kleingruppe angstfrei mitzuarbeiten und durch eigene Beiträge Selbstsicherheit zu erlangen. Lernumwege und fehlerhafte Ansätze, die in Sackgassen führen, sind nicht vergebens, sondern schärfen das zukünftige Problemlöseverhalten. Einzelkämpfer müssen ihre Fähigkeiten in den Dienst der Gruppe stellen, um den Gemeinschaftserfolg zu sichern. Probleme entstehen dann, wenn Spezialisten ganze Arbeitsbereiche an sich reißen und den Gruppenrest abhängen. POLOCZEK¹ hält daher,

¹ POLOCZEK, J. (2002) S. 5

eine organisierte Gruppenarbeit mit wechselnden Aufgabenstellungen...[für]...hilfreich, um die koordinierte Arbeit aufrecht zu erhalten.

Deshalb werden zusätzliche Teamstrukturen eingeführt. Sie werden in einem rotierenden Verfahren belegt. Im Handbuch¹ zum LEGO-MINDSTORMS Baukasten für Schulen finden sich Vorschläge zu den Rollen der einzelnen Teammitglieder. Die Autoren gehen von größeren Lerngruppen aus und teilen jedes Team in Konstrukteure oder Programmierer auf. Zusätzlich erhalten die Gruppenmitglieder weitere Aufgaben als Teammanager, Kommunikationsspezialist, Materialspezialist und Informationsspezialist. Die Aufgabenbereiche der Spezialisten sind in einem Arbeitsblatt erläutert. Eine vergleichbare Form teamspezifischer Aufgaben wird auch von BAUMANN² formuliert.

An der Musterschule kann aufgrund der zahlreich zur Verfügung stehenden Baukästen die Gruppengröße auf 3 – 4 Personen beschränkt sein. Diese Schüleranzahl ermöglicht zum einen bereits eine teamworkorientierte, zeitökonomische und arbeitsteilige Vorgehensweise. Zum anderen kann das Team aber auch Aufgaben gemeinsam durchführen. Bei 17" Bildschirmen können z.B. bis zu 3 Personen gemeinsam an einem PC-Arbeitsplatz arbeiten. Die Vorschläge aus dem Handbuch wurden deshalb für 3-Personen-Gruppen etwas modifiziert: Die Rolle des Kommunikationsspezialisten wurde gestrichen, seine Aufgaben wurden dem Informationsspezialisten übertragen. Wichtig erscheint mir auch, dass Konstrukteure und Programmierer sich gegenseitig abwechseln, damit jedes Gruppenmitglied Kenntnisse und Fähigkeiten auf beiden Gebieten erwirbt.

Selbstverständlich hat auch die weniger team-orientierte Sozialform „Unterricht im Klassenverband“ mit ihren Ausprägungen „Frontalunterricht“ und „fragend-entwickelnder Unterricht“ ihre Berechtigung im Informatikunterricht. Diese eignen sich für Einführungsphasen zur Einstimmung auf ein neues Unterrichtsthema, zur Rückmeldung über Vorkenntnisse der Schülerinnen und Schüler, sowie zum Sammeln und Darstellen von Ergebnissen im Plenum. Als Sitzordnung für diese Unterrichtsabschnitte hat sich der Innenkreis ohne PC bewährt.

Die Unterrichtsstrukturierung erfolgt durch Arbeitsblätter. Sie haben auffordernden Charakter und lassen ein individuelles Arbeitstempo in den Gruppen zu (vgl.³). Durch Zusatzaufgaben ist die Möglichkeit zur Binnendifferenzierung gegeben. Anhänge versorgen die Gruppen mit Informationen und Zeitplänen. Alle Materialien werden in einem Schnellhefter gesammelt. Die Arbeitsblätter beinhalten eine „Erledigt“-Spalte zur Protokollführung und zum Leistungsnachweis. Fahrzeugbeschreibungen charakterisieren die Roboter (Instanzen) durch Eigenschaften (Attribute) und Fähigkeiten (Methoden).

¹ LEGO MINDSTORMS für SCHULEN (2000), S. 46 - 47

² BAUMANN, R. (1996), S. 192

³ HUBWIESER, P. (2001), S. 113

6. DIE UNTERRICHTSREIHE

Ein vollständiger Entwurf zu jeder Doppelstunde würde den Rahmen dieser Arbeit sprengen, zumal die Didaktik zur Unterrichtsreihe bereits in Kapitel 3 dargelegt wurde. Die didaktischen Bemerkungen zu den einzelnen Stunden beschränken sich deshalb auf einige Kernpunkte, die mir wichtig erscheinen. Lernziele und Stundenverlauf werden tabellarisch aufgeführt. Auf die Unterrichtsmethoden wird nur dann eingegangen, wenn von der allgemeinen Form Gruppenarbeit, wie sie in Kapitel 5 beschrieben und begründet worden ist, abgewichen wird. Nicht alle Entwürfe konnten aus Zeitgründen durchgeführt und erprobt werden. Im Anwendungsfall sind den Planungen jeweils Bemerkungen zum Verlauf beigelegt.

6.1 DIE ERSTE DOPPELSTUNDE

Zu Beginn der Unterrichtsreihe Lego-MINDSTORMS im WP-Unterricht müssen im Plenum die Erwartungen an Projektarbeit, Vorgehensweise, Zielsetzung, Zeitvorgaben, Protokolle und Gruppenstärke geklärt werden. Abhängig von Alter und Erfahrungsstand der Schülerinnen und Schüler bieten sich als Methoden ein Brainstorming, ein fragend-entwickelndes Unterrichtsgespräch oder ein Lehrervortrag an. Fixiert man die Vereinbarungen auf einem Flip-Chart, so können diese über die gesamten Projektzeitraum sichtbar im Raum aufgehängt werden.

Anschließend bilden sich die Teams. Bis auf die Gruppengröße (am besten 3, maximal 4 Personen) soll es hierbei keine einschränkenden Vorgaben geben.

Die Einführungsphase soll weiterhin die Schülerinnen und Schüler für das neue Unternehmen gewinnen. Ein dekorierter Roverbot, wie er als Ergebnis der ersten Stunden konstruiert werden soll, wird zur intrinsischen Motivation verwendet. Er führt einfache Bewegungsabläufe auf einem Aktionsfeld durch. Der ROVERBOT ist das Basisfahrzeug für die Einführungslektionen.



Abbildung: Dekorierter ROVERBOT mit Flügel und beweglichen Klauen

Die Einführungslektion bis zur Bauphase des ROVERBOTS dauert ca. 15 Minuten. IR-Tower, RCX mit (geladenen!) Akkus, 2 Motoren, 2 Druck-, sowie ein Lichtsensor und 4 kurze Verbindungskabel sollten für die Arbeitsgruppen bereit liegen. Wenn ein Batteriewechsel stattgefunden hat, muss gegebenenfalls die „Firmware“ erneut geladen werden. Dieser Vorgang dauert ca. weitere 5 Minuten. Zur Aufbewahrung der Teile haben sich stapelbare (IKEA)-Plastikkisten bewährt, die mit dem Gruppennamen versehen werden. Die Boxen dienen auch als „Garage“ für die späteren Fahrzeuge.

In der ersten Bauphase konstruieren alle Gruppen den gleichen Robotertyp, wie er in der Constructopedia¹ vorgestellt wird. Er soll wie im Einführungsvideo mit Reifen bestückt sein. Die Beschränkung auf einen Fahrzeugtyp hat für die Einführungsphase mehrere Vorteile. Die Konstruktion ist sehr einfach; der Bau eines Kettenantriebs wäre z.B. zeitlich und mechanisch deutlich aufwändiger. Weiterhin sind die vorgefertigten Programme der Trainingslektionen auf den Roverbot abgestimmt. Andere Fahrzeuge haben ein anderes Fahrverhalten, die Videoanimationen könnten so nur bedingt zum Vergleich herangezogen werden. Da die Grundkonstruktion des Antriebsblocks in sehr vielen Fahrzeugtypen Verwendung finden kann, stellt der kollektive Bau eines Robotertyps („ROVERBOT“) kein Hindernis für eine spätere Individualisierung dar.



Abbildung: Erste Bauphase mit Lego MindStorms

Lernziele: Die Schülerinnen und Schüler sollen

- durch die Einführung für ihre weitere Arbeit an diesem Thema motiviert werden
- die Regeln für Gruppenarbeit erarbeiten und verinnerlichen
- das Robotics-Invention-System 2.0 kennen lernen
- die Hard- und Software für das Robotics-Invention-System 2.0 konfigurieren können
- mit Hilfe der Constructopedia einen ROVERBOT bauen können
- eine Steuerung der Motoren programmieren können
- Programme laden, übertragen, ändern und speichern können
- ihre Rolle im Team finden und festlegen
- durch das Arbeiten in Gruppen soziale Kompetenzen erwerben

¹ LEGO MINDSTORMS Constructopedia RIS 2.0, S. 10 - 35



Abbildungen: Arbeiten mit der Constructopedia.



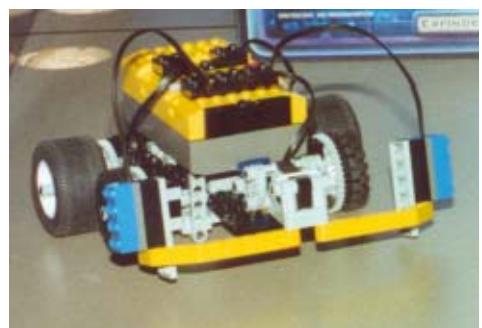
Ein ROVERBOT als Basisfahrzeug.

Geplanter Unterrichtsverlauf:

Phase	Inhalt	U-Form	Medien
Einführung	Vorstellung Gruppenarbeit und Lego-MINDSTORMS Baukasten,	Brainstorming Lehrervortrag	Flip-Chart Lego-MINDSTORMS Baukasten, dekorierter ROVERBOT
	Gruppenbildung	S-S-Interaktion	---
Erarbeitung	Hardware Konfiguration	Gruppenarbeit	Arbeitsblatt 1, PC, IR-Tower
Einführung	Interaktive Einführungstour Übertragen der „Firmware“	Videosequenz	RIS 2.0 Software, RCX, Motoren, Sensoren
Erarbeitung	Bau eines ROVERBOTS	Gruppenarbeit	Lego-MINDSTORMS Baukasten
Sicherung	Testen des ROVERBOTS	Gruppenarbeit	RIS 2.0 Software, ROVERBOT
Einführung	Programmierung	Videosequenz	RIS 2.0 Software, ROVERBOT
	Festlegung der Rollen im Team	Gruppensitzung	Anlage Arbeitsblatt 1
Sicherung	Umbauten am ROVERBOT	Gruppenarbeit	Lego-MINDSTORMS Baukasten, ROVERBOT

Die Bildung der Arbeitsgruppen erfolgte ohne Probleme. Einige wenige Gruppen bestanden aus Kapazitätsgründen aus vier Schülerinnen und Schülern. Bei diesen Gruppen sollte im weiteren Verlauf besonders auf eine arbeitsteilige Vorgehensweise geachtet werden, um Leerlaufphasen zu vermeiden.

Abbildung rechts: ROVERBOT mit Einzelstoßfänger und 2 Lichtsensoren außen



Die Aussicht, selbst Roboter bauen zu dürfen, und der fertige ROVERBOT waren genug Motivation, um mit Begeisterung die Gruppenarbeit zu beginnen. Erfahrungen im Umgang mit Lego-Bausteinen sind auch heute noch bei den meisten Schülerinnen und Schülern aus Kindheitstagen vorhanden. Bei Problemen half die Constructopedia weiter. Die Arbeitsfreude war in manchen Gruppen so groß, dass ein Hitzefrei in der 6. Unterrichtsstunde ignoriert wurde! Am Ende der ersten Doppelstunde standen daher bereits die ersten fahrfähigen ROVERBOTS zu Verfügung. Mit Überraschung registrierte ich die Faszination der Schülerinnen und Schüler, die von der Technik ausging. Vielleicht gelingt es damit, zu physikalischen Themen wie Drehmomenten und Übersetzungsverhältnissen einen anwendungsorientierten und fachübergreifenden Zugang zu entwickeln.



Abbildungen: Bilder vom Roboterbau WPU-Kurse
Jahrgangsstufe 9 und 10

6.2 DIE ZWEITE DOPPELSTUNDE

Im Vordergrund der zweiten Doppelstunde steht das Kennenlernen und Verstehen der Lego-MindStorms Software RIS 2.0. Die Schülerinnen und Schüler erlernen den Umgang mit der Benutzeroberfläche. In Trainingslektionen werden zentrale Elemente der Programmierung vorgestellt. Übungsobjekt ist stets der ROVERBOT aus der ersten Stunde. Lerngruppen mit Programmierkenntnissen werden diese Abschnitte zügig bearbeiten können, da ihnen übergeordnete Programmstrukturen bekannt sind. Sie müssen lediglich die speziellen Steuerungsbefehle für Motoren und Töne erlernen und sich an das Übertragen der Programme gewöhnen. Für den heutigen Übungsausschnitt (Lektionen 3 – 6) benötigen die Arbeitsgruppen voraussichtlich 35 Minuten, so dass dieser Unterrichtsabschnitt zusammen mit der Anfangsphase die erste Stunde in Anspruch nehmen wird.

Eine Konstruktionsphase sorgt zu Beginn der zweiten Hälfte für einen Phasenwechsel. Mit der angefertigten Fernbedienung kann eine Handsteuerung für den Roverbot implementiert werden. Die beiden Beispielprogramme „S_SENSOR.LSC“ und „G_SENSOR.LSC“ sind nicht Bestandteil der Originalsoftware. Sie befinden sich auf der beigefügten CD und Diskette und müssen den Benutzern zur Verfügung gestellt werden.

Das Programm „S_SENSOR“ ähnelt einem Programm aus den Trainingslektionen. Der (Unter-)Programmteil, der bei betätigtem Sensor (logisch „1“) ablaufen soll, wird seitlich neben dem „normalen“ (Haupt-)Programmcode platziert. Das Hauptprogramm wird sequentiell durchlaufen. Es wird jedoch immer dann sofort unterbrochen, wenn das Ereignis Sensor aktiviert wird, um das Unterprogramm zu starten. Am Unterprogrammende wird zur Sprungstelle im Hauptprogramm zurückgekehrt.



Abbildung: Programmcode S_SENSOR.lsc



Das Springen in Programmen macht den Programmablauf unübersichtlich und gehört der (BASIC-)Vergangenheit an, zumal beim unkommentierten Springen das Programmverhalten nicht unmittelbar aus der Programmstruktur zu erkennen ist. Im Sinne eines imperativen Programmierstils ist es günstiger, Schleifen- oder Auswahlstrukturen zu verwenden. Dadurch werden Programme lesbarer und nachvollziehbar gestaltet. Die Variante „G_SENSOR“ zeigt eine solche Möglichkeit auf, indem sie auf eine zweiseitige Auswahl (IF-Else-Abfrage) zurückgreift.

Abbildung links: Programmcode G_SENSOR.lsc

Die Vorteile der implementierten zweiseitigen Abfragen werden in Aufgabe 6 deutlich. Die Schülerinnen und Schüler müssen eine zweifach geschachtelte IF-Else-Abfrage verwenden, damit alle Tastenvariationen der Fernbedienung berücksichtigt werden. Eine Lösungsmöglichkeit ist auf der CD unter dem Programmnamen „FERNBEDIENUNG.LSC“ zu finden.

Abbildung rechts: AUFGABE 6
Programmcode FERNBEDIENUNG.lsc



Abbildung:
Kommentarblock

Kommentare erhöhen die Lesbarkeit von Programmen. Auf der Programmieroberfläche der RIS 2.0 Software können über die Optionen „BEARBEITEN“ → „KOMMENTAR“ die grauen Kommentarbausteine eingefügt werden. Etwas umständlich ist die Handhabung, da der Kommentartext erst dann lesbar und beschreibbar ist, wenn die Registerkarte angeklickt worden ist.

Unter dem Menüpunkt „KLEINE BLÖCKE“ → „ERWEITERT“ ist das Programmende-Zeichen schwer aufzufinden. Da es für das Funktionieren der Programme nicht notwendig ist, wurde es von den Autoren leider ziemlich versteckt angeordnet. Trotzdem sollten die Schülerinnen und Schüler ihre Programme mit diesem Zeichen abschließen. Bereits bei etwas längeren Programmen reicht die Bildschirmhöhe nicht für den vollständigen Code aus. Beim Scrollen werden leicht Programmreste übersehen. Eine Endmarkierung ist daher sehr hilfreich.



Abbildung
Programmende-Zeichen

In der zweiten Doppelstunde sollen die Arbeitsgruppen noch nicht arbeitsteilig vorgehen, sondern gemeinsam Arbeitsblatt 2 bearbeiten. Die anfänglichen Trainingslektionen sind von grundlegender Bedeutung für Programmierer wie auch für Konstrukteure und müssen daher von allen Schülerinnen und Schülern absolviert werden. Die Programmierelemente Kommentarblock und Programmende-Zeichen lassen sich am besten mit einem Beamer im Plenum demonstrieren. Falls kein Beamer verfügbar ist, kann mit Hilfe einer pädagogischen Software wie MasterEye® der Lehrerbildschirm im Demobetrieb auf alle Schülerbildschirme übertragen werden.

Lernziele:

Die Schülerinnen und Schüler sollen

- ihre Rolle im Team ausfüllen
- mit den Trainingslektionen folgende Programmstrukturen und –befehle kennen und anwenden lernen:
 - Programme laden, übertragen, ändern und speichern
 - die Hilfefunktion der Software RIS 2.0
 - die Registerkarten der Programmblöcke
 - mit dem Befehlsassistenten die Registerkarten bearbeiten
 - Editorfunktionen: Kopieren, Scrollen und Zoomen
 - die Funktionsweise des Berührungssensors
 - die Funktionsweise des Lichtsensors
 - den Schleifenbefehl „WIEDERHOLEN“
 - die If-Abfrage „WARTE BIS“
 - die If-Else-Abfrage „JA ODER NEIN“
 - die Timerfunktion
- eine Fernbedienung nach Anleitung konstruieren können
- die Programme G_Sensor und S_Sensor analysieren und beurteilen können
- Kommentarblock, Programmende-Zeichen und Logbuch kennen lernen
- ein Fernbedienungsprogramm für einen Roverbot programmieren können
- Zustandsdiagramme für die Akteure erstellen können

Geplanter Unterrichtsverlauf:

Phase	Inhalt	U-Form	Medien
Einführung	Rückblick auf Einführungsstunde	Lehrer- und Schülervortrag	Erste ROVERBOTS
	Die Spezialisten erledigen ihre Aufgaben	Gruppenarbeit Arbeitsteilig	Arbeitsblatt 2
Erarbeitung	Bearbeiten der Trainingslektionen 3 – 6	Videosequenz	RIS 2.0 Software, PC, RCX, ROVERBOT mit Sensoren
	Bau einer Fernbedienung	Gruppenarbeit im Team	Lego-MINDSTORMS Baukasten
	Analyse von Programmen		S_Sensor und G_Sensor
Sicherung	Programmierstil Block Kommentar einfügen Programmende-Zeichen	Plenum	Beamer oder MasterEye
Erarbeitung	Programmieren einer Fernsteuerung Erstellen von Zustandsdiagrammen	Gruppenarbeit im Team	RIS 2.0 Software, PC, RCX, ROVERBOT mit Fernbedienung

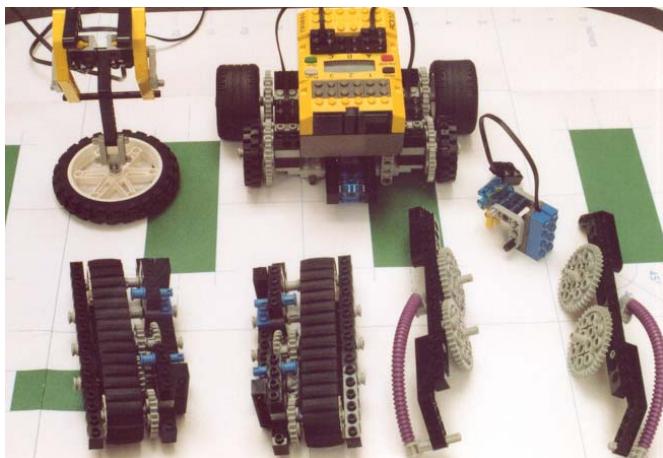


Abbildung: ROVERBOT mit

- Fernbedienung
- Kettenantrieb,
- „Beine“
- Lichtsensor

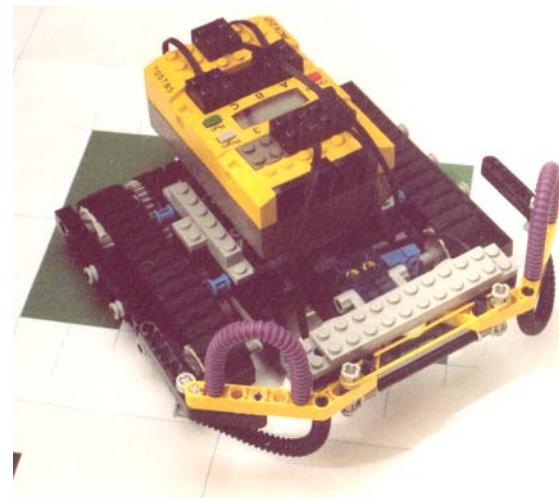


Abbildung: ROVERBOT mit
Doppelstoßfänger

6.3 DIE DRITTE DOPPELSTUNDE

Das Hauptgewicht der dritten Doppelstunde liegt im Festigen und Erweitern der Programmierfähigkeiten. Die Erfahrungen aus der letzten Doppelstunde zeigten, dass zu lange Trainingslektionen die Motivation negativ beeinflussen. Die heutige Doppelstunde soll daher stärker gegliedert und abwechslungsreicher sein. Weiterhin soll das eigenständige Lösen von Problemen durch Modellieren und Programmieren gefördert werden. Bevor die Schülerinnen und Schüler die restlichen Trainingslektionen bearbeiten, ist deshalb eine Konsolidierungsphase vorgesehen. Eine Aufgabenteilung zwischen Programmierern und Konstrukteuren findet erst zum Ende der Stunde statt, denn die heutigen Einführungslektionen sind zum allgemeinen Verständnis der beiden Komponenten, Konstruktion und Programmierung, zu elementar und zu unverzichtbar, als dass bereits zu Anfang eine Spezialisierung stattfinden könnte.

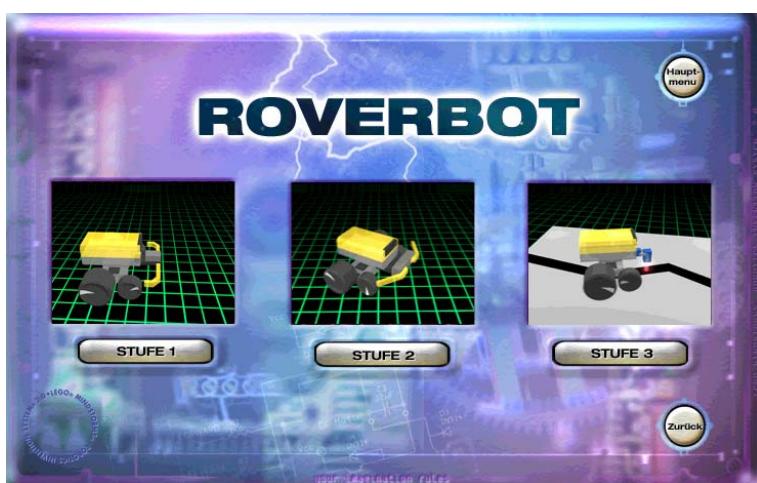


Abbildung: Übungsaufgaben „ROVERBOT“

Videosequenz erläutert. Eine parallele Struktur von 12 Aufgaben in 3 Schwierigkeitsgraden gibt es auch für die beiden anderen Robotertypen „ACROBOT“ und „INVENTORBOT“. Alle Übungsaufgaben sind mit Programmbeispielen versehen. Leider verführt diese Option die Schülerinnen und Schüler sehr schnell zum Nachschauen. Der pädagogische Effekt, selbständig Problemsituationen zu analysieren, dazu passende Programme zu entwerfen und diese erfolgreich umzusetzen, wird dadurch nicht unterstützt. Aus didaktischen Gründen habe ich daher die Inhalte einiger Programmbeispiele verändert. Die Übungsprogramme der Trainingsaufgaben für den Roverbot sind bei einer Standardinstallation der RIS 2.0 Software unter dem Pfad:

„C:\PROGRAMME\LEGO MINDSTORMS\RIS 2.0\DATA\RisMODULES\CHALLENGES\001_ROVERBOT“

abgespeichert. Mit Hilfe des Tools „TAUSCHEN.BAT“ können die modifizierten Programme von der beigefügten Diskette, Verzeichnis „A:\NEW_ROVERBOT“, in das Originalverzeichnis kopiert werden. Der Schreibschutz der ursprünglichen Programmcodes wird zuvor aufgehoben und anschließend wieder gesetzt. Zur Archivierung der überschriebenen Programme ist auf der Diskette ein Verzeichnis „OLD_ROVERBOT“ eingerichtet. Sie können jederzeit durch „R_TAUSCHEN.BAT“ zurückgeholt werden. Die geänderten Programmcodes sind im Anhang des 3. Arbeitsblattes wiedergegeben. Sie betreffen folgende Programme:

Im Untermenü „LEKTIONEN“ können außer „TRAININGSELEKTIONEN“ auch „AUFGABEN“ bearbeitet werden. Sie sind in 3 Gruppen („STUFE 1 – 3“) von jeweils 4 Aufgaben zusammengefasst. Für eine Aufgabengruppe ist jeweils ein mechanischer Aufbau mit Sensoren notwendig. Gleichzeitig steigt der Schwierigkeitsgrad von Aufgabe zu Aufgabe und von Stufe zu Stufe an. In jedem Abschnitt wird die erste Aufgabe per

„HAUPTMENÜ“ → „LEKTIONEN“ → „AUFGABEN“

„Stufe 1:“	„GUARDBOT“ „DANCER“ „BOOMERANG“ „PUSHBOT“	(*) (*) (*)	nur Schleifenvorgabe nur Ja-oder-Nein-Abfrage ergänzte Originalversion keine Vorgaben
„Stufe 2:“	„EXPLORER“ „MOZART“ „LIGHT SEEKER“ „WALL HUGGER“	(*) (*) (*)	keine Vorgaben neue, unvollständige Programmvariante nur Schleifenvorgabe nur Schleifenvorgabe
„Stufe 3:“	„LINE FOLLOWER“ „CLEAN SWEEP“ „LIGHT DRIVE“ „SOUND MAKER“	(*) (*) (*)	neue, unvollständige Programmvariante keine Vorgabe ergänzte Originalversion ergänzte Originalversion

Die mit (*) versehenen Aufgaben entsprechen dem Kenntnisstand der Schülerinnen und Schüler. Sie können auch ohne vorherige Behandlung der „TRAININGSLEKTIONEN FÜR FORTGESCHRITTENE“ bearbeitet werden. In Arbeitsblatt 3 werden die Schülerinnen und Schüler aufgefordert, sich mit einigen der Aufgaben zu befassen.

Die folgenden Trainingslektionen 7 – 9 erschließen dem fortgeschrittenen Anwender spezielle Features der Programmiersprache RIS 2.0. Wurden bisher vom Robotertyp abhängige „GROSSE BLÖCKE“ als Methoden zur Steuerung der Ausgänge verwendet, so erhalten die Schüler und Schülerinnen jetzt mit den „KLEINEN BLÖCKEN“ eine direkte Kontrolle über Motoren und Lampen, sowie über Töne, Meldungen und Variable. Sie erkennen den prozeduralen Charakter der „GROßEN BLÖCKE“, die aus zahlreichen „KLEINEN BLÖCKEN“ zusammengesetzt sind. Am Beispiel des Großen Blocks „VORWÄRTS“ werden einige der „KLEINEN BLÖCKE“ aus dem Bereich „LEISTUNG“ erläutert.

Eine Neuerung stellen die „KLEINEN BLÖCKE“ aus dem Behälter „Komm.“(unikation) dar. Stellvertretend für alle Kommunikationsbefehle wird der Befehl „WERT ANZEIGEN“ demonstriert. Er ist wichtig, um über das Display des RCX Informationen wiedergeben zu können. Mit weiteren Befehlen aus dem Behälter Kommunikation können auch zwei RCX untereinander kommunizieren. Auf diese Weise lassen sich in komplizierten Robotern mehrere RCX-Bausteine miteinander koppeln, um z.B. mehr als 3 Motoren ansteuern zu können (vgl.¹).

Der „KLEINE BLOCK“ „VARIABLE“ ermöglicht das Deklarieren einiger Variablen. Sie können Dezimalzahlen im Bereich zwischen –3276,8 und +3276,7 mit einer Dezimalstelle Genauigkeit speichern. Dies entspricht einem 2 Byte-Wort. Die Verfügbarkeit von Variablen erweitert die Programmierungsmöglichkeiten erheblich. Es können dadurch „lernfähige“ Roboter programmiert werden, die z.B. nach einer bestimmten Anzahl von Fehlversuchen eigenständig die Strategie wechseln. Hilfreich ist in diesem Zusammenhang auch die Möglichkeit, mit einem Zufallsgenerator Variablenwerte zu definieren. Beim Ändern von Variablenwerten durch mathematische Operationen müssen die Bereichsgrenzen berücksichtigt werden, da eine Overflow-Kontrolle nicht stattfindet.

¹ LOUIS, D. & MÜLLER, P. (2001), S. 123 - 128

Auf die restlichen „KLEINEN BLÖCKE“ „SOUND“, „ZURÜCKS.“(etzen) und „ERWEITERT“ wird in den Trainingslektionen nicht eingegangen. Ihre Inhalte sind zum Teil selbsterklärend („SOUND“) oder können mit Hilfe von Beispielprogrammen oder der Direkthilfe verstanden werden.

Der Behälter „EIGENE BLÖCKE“ dient zur Aufnahme von „GROßen BLÖCKEN“ aus eigener Konstruktion. Sich wiederholende Kommandos und Kontrollstrukturen können auf diese Weise in „EIGENEN BLÖCKEN“ zusammengefasst werden. „EIGENE BLÖCKE“ entsprechen den Prozeduren aus imperativen Programmiersprachen. Sie vereinfachen die Programmstruktur erheblich und dienen der Programmierkultur. Da die Symbole der Programmbausteine der Lego-Software recht groß ausfallen, sind nur wenige bei normalem Zoomfaktor auf der Bildschirmoberfläche zu überschauen. Ein kleinerer Zoomfaktor vermindert jedoch die Lesbarkeit; beim Hin- und Her-Scrollen verliert man leicht den Überblick. Daher sollten die Schülerinnen und Schüler sollten zum Entwickeln „EIGENER BLÖCKE“ angehalten werden. Aufgabe 4 und 5 des dritten Arbeitsblattes enthalten daher Aufträge zum Programmieren „EIGENER BLÖCKE“.

Lernziele: Die Schülerinnen und Schüler sollen

- die Problemstellungen der Übungsaufgaben
 - analysieren können
 - mechanisch-konstruktiv umsetzen können
 - einen Programmcode formulieren und testen können
- mit den Trainingslektionen
 - den Aufbau „GROßer BLÖCKE“ kennen und verändern lernen
 - die „KLEINEN BLÖCKE“ als elementare Steuerelemente kennen und anwenden lernen
 - „EIGENE BLÖCKE“ deklarieren und gestalten können
 - VARIABLEN benennen und deklarieren können
- Zustandsdiagramme und Programmcodes
 - zu den „EIGENEN BLÖCKE DREHENLINKS“ und „DREHENRECHTS“ erstellen können
- den Roverbot in Antrieb und Sensorbestückung sowie in Dekoration und zusätzlichen musikalischen Aktionen modifizieren können

Geplanter Unterrichtsverlauf:

Phase	Inhalt	U-Form	Medien
Sicherung	Problemstellungen mit ROVERBOTS Lösen von Trainingsaufgaben Analyse von Programmen	Gruppenarbeit Im Team	Arbeitsblatt 3
Erarbeitung	Bearbeiten der Trainingslektionen 7-9	Videosequenz	RIS 2.0 Software, PC, RCX, ROVERBOTS mit Sensoren
Sicherung	Zustandsdiagramm und Programmcode „EIGENE BLÖCKE“	Gruppenarbeit Im Team	
Erarbeitung Sicherung	Konstruktive Änderungen Programm ergänzungen	Gruppenarbeit Arbeitsteilig	Protokoll



Abbildungen: ROVERBOT mit Lichtsensor und großen Einzelstoßfänger

6.4 WEITERE ÜBUNGSSTUNDEN

In Abhängigkeit von den Fähig- und Fertigkeiten der Lerngruppe empfiehlt sich für die nächsten Stunden ein Projekt „Dosenwettstreit“. Bei Informatikanfängern muss gegebenenfalls zuvor noch eine Doppelstunde mit Übungen in den Bereichen „KLEINE BLÖCKE“, „VARIABLE“ und „EIGENE BLÖCKE“ durchgeführt werden. Bei erfahreneren Lerngruppen, die wie im vorliegenden Fall bereits über Programmierkenntnisse verfügen, sollte dies nicht notwendig sein.

Aufgabenstellungen für zusätzliche Übungsstunden:

Der Roboter soll

- (mehrfach) eine symmetrische Acht fahren
- langsam anfahren und bremsen bzw. ausrollen können
- auf der Übungsfläche beim Anstoßen
 - zufällig wenden
 - die Anstöße zählen und anzeigen
- vorn und hinten mit einem Einzelstoßfänger ausgestattet sein und beim Anstoßen die Fahrtrichtung wechseln
- Farben an der unterschiedlichen Helligkeit erkennen und durch verschiedene Töne oder Richtungsänderungen reagieren
- mit einem Filzstift versehen, geometrische Figuren zeichnen und dabei die Ecken zählen und anzeigen (siehe auch „MALBOT“ bei „LEKTIONEN“ → „PRO-AUFGABEN“)
- umgebaut werden (Konstrukteure) und über andere Fähigkeiten (Programmierer) verfügen (siehe auch „ACROBOT“ bei den „LEKTIONEN“ → „AUFGABEN“)

6.5 DER DOSENWETTSTREIT

Der Dosenwettstreit kann das Finale einer Unterrichtsreihe mit Lego MINDSTORMS darstellen. In einem abschließenden Wettkampf zeigen alle Schülerinnen und Schüler ihre erworbenen Kenntnisse und Fähigkeiten, sowohl als Einzelperson, als auch als Team. Die Gruppenarbeit kann auch zur Leistungsbewertung herangezogen werden.

Die Idee zum Wettstreit der Arbeitsgruppen stammt aus dem HANDBUCH LEGO MINDSTORMS FÜR SCHULEN¹, LÖSEN EINER TEAMAUFGABE MIT DEM LEGO SET #9793/9794. Die Autoren gehen dabei von einer Lerngruppe aus, die sich im Rahmen einer Projektwoche (8 - 10 Tage) ausschließlich mit Lego MindStorms befasst. Ich halte diesen Zeitrahmen für sehr großzügig bemessen. Meine Planungen sehen eine zügigere Vorgehensweise im Rahmen des 2-stündigen WP-Unterrichts vor. Mit den entsprechenden Vorkenntnissen sollten die Schülerinnen und Schüler die Vorbereitungen zum Wettstreit in 3 – 4 Doppelstunden bewerkstelligen können. Die Tätigkeitslisten aus dem Handbuch³⁰ habe ich den neuen Modalitäten angepasst. Sie werden den Schülerinnen und Schülern als Anhang zum 4. Arbeitsblatt zur Verfügung gestellt. Um diesen Zeitplan einhalten zu können, soll in den Teams arbeitsteilig vorgegangen werden. Zur Abstimmung der Arbeitsschritte finden zu Beginn und am Ende einer Unterrichtsstunde regelmäßige Teamsitzungen statt. Die Sitzungen leitet der Teammanager; Protokoll führt der Informationsspezialist (4-er Gruppen: Kommunikationsspezialist). Alle Informationsspezialisten informieren in jeder Doppelstunde den Lehrer über den aktuellen Arbeitsstand. Die Funktionen Teammanager, Informations-, (Kommunikations-) und Materialspezialist rotieren während des Projekts „Dosenwettstreit“ innerhalb einer Arbeitsgruppe.

Die Arbeitspläne sind so konzipiert, dass sowohl Programmierer, als auch Konstrukteure sofort ihre Aufgaben erfüllen können.

Im Gegensatz zu den früheren Aufgaben stellt der Dosenwettstreit eine größere Herausforderung dar. Im Bereich der Trainingslektionen waren die Übungen in der Einführungsphase überschaubar und sehr konkret auf die jeweilige Lektion bezogen formuliert. In der Konsolidierungsphase wurden die Aufgabenstellungen zwar komplexer und die Programme damit umfangreicher, jedoch gab es immer Vorgaben zu Sensoren oder Antriebstechnik. Diese Einschränkungen waren didaktisch sinnvoll, um so schrittweise die Möglichkeiten des RIS 2.0 kennen zu lernen.

Im Dosenwettstreit haben die Teilnehmer alle Entwicklungsfreiheiten. Es gibt nur Vorgaben über Zeitdauer des Wettkampfes und Größe des Roboters. Die Entscheidung über die Roboter, mit der die Teams versuchen, die Dosen aus dem Spielfeld zu räumen, bleibt ihnen überlassen. Weiterhin muss eine Strategie entwickelt werden, das Spielfeld möglichst effektiv nach Dosen abzusuchen. Die diesbezüglich von der Gruppe getroffenen Entscheidungen werden von den Konstrukteuren und Programmierern umgesetzt. Damit durchlaufen die Schülerinnen und Schüler den gesamten Zyklus des Software Engineerings.

¹ LEGO MINDSTORMS FÜR SCHULEN:
Lösen einer Teamaufgabe mit dem Lego Set #9793/9794, S. 41 – 69

Lernziele: Die Schülerinnen und Schüler sollen

- die Problemstellungen der Teamaufgabe
 - analysieren und modelllieren können
 - sich auf eine Strategie und Konstruktion einigen können
- als Konstrukteur
 - die Vorgaben mechanisch-konstruktiv umsetzen und optimieren können
 - das Fahrzeug optisch attraktiv gestalten können
- als Programmierer
 - Zustandsdiagramme und Algorithmen zum Abscannen des Spielfeldes zum Entfernen der Dosen
 - Aktionsmelodien entwickeln und implementieren können
- In einem Team aus Konstrukteuren und Programmierern
 - konstruktiv miteinander und
 - effektiv arbeitsteilig vorgehen können
- Testphasen zur Optimierung der Strategie und Konstruktion verwenden können
- Verantwortung für die übertragenen (Teil-)Aufgaben übernehmen können
- Zeitpläne einhalten können
- eine Dokumentationen zum Projekt erstellen können

6.6 MÖGLICHKEITEN ZUR FORTSETZUNG DER UNTERRICHTSREIHE

Selbstverständlich kann die Unterrichtsreihe auch fortgeführt werden. Waren alle Objekte bisher Roboter, die mobil waren, so könnte eine Fortsetzung sich mit ortsgebundenen Robotern beschäftigen. Im Sinne der Objektorientierung werden der Klasse „Roboter“ die beiden Unterklassen „Mobile_Roboter“ und „Ortsfester_Roboter“ zugeordnet. Ideen zur Konstruktion und Aufgabenstellung ortsgebundener Roboter sind u.a. in der RIS 2.0 Software unter „LEKTION“ → „PRO-AUFGABEN“ zu finden. In der Constructopedia des Exploration Mars Sets wird ein Mars-Lander beschrieben, der mit einem beweglichen Roboterarm versehen werden kann.

Aber auch im Bereich der mobilen Roboter bieten sich Fortsetzungsmöglichkeiten an. Mit den zusätzlichen Bauteilen des Exploration Mars Sets kann ein gelände-gängiges Erkundungsfahrzeug mit 3 Achsen und Schneckengetriebe konstruiert werden. Ein vergleichbarer Robotertyp lässt sich auch mit konventionellen Bauteilen aus den RIS 2.0 Kästen herstellen.

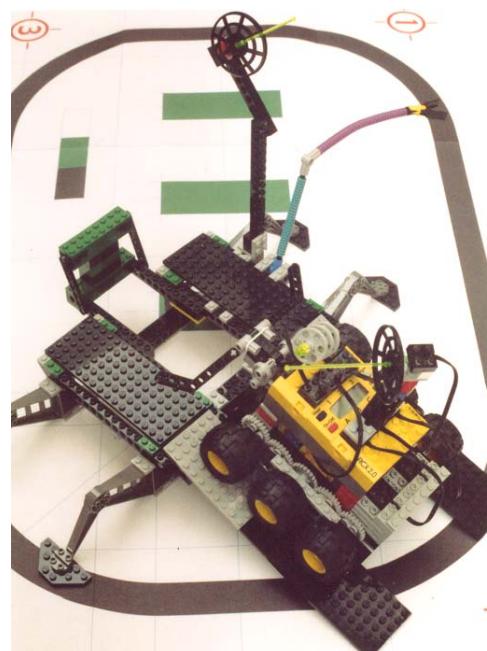
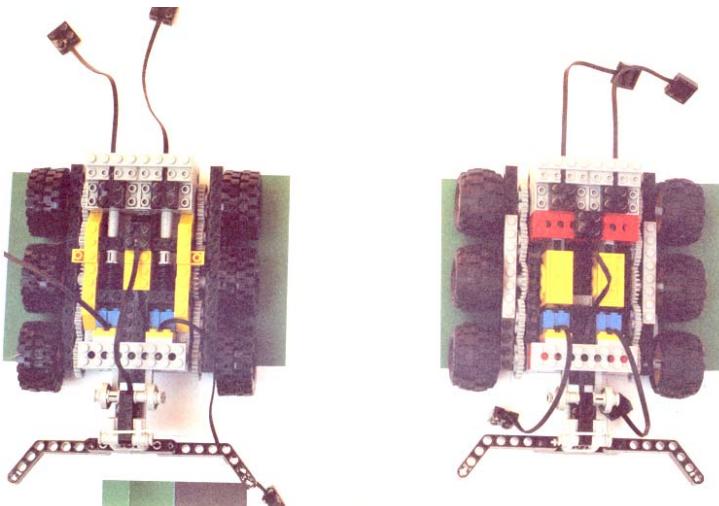


Abbildung: Marsroboter auf dem Mars-Lander



Abbildungen:

Mars-Exploration Set (rechts)

Konventionelle Bauart RIS 2.0 (links)
mit offenem Schneckengetriebe und
Doppelreifen

2 zentrale Lichtsensoren

Ein anderer Typ von mobilen Robotern orientiert sich am klassischen Kraftfahrzeug. Er verfügt nur über einen Antriebsmotor. Die Übertragung der Antriebskraft auf die (Hinter-) Räder erfolgt mit einem Differentialgetriebe, um Längenunterschiede bei Kurvenfahrten zu kompensieren. Die Vorderräder sind schwenkbar angeordnet, so dass Richtungsänderungen über eine Zahnstangenlenkung möglich sind. Diese wird von einem zweiten Motor in Verbindung mit einem stark untergesetzten Schneckengetriebe betätigt. Um die Geradeaus-Stellung eindeutig zu definieren, bietet sich der Rotationssensor an. Er zählt, am Schneckengetriebe der Lenkung montiert, die Anzahl der Umdrehungen nach links (-) oder rechts (+). Der Variablenwert 0 entspricht dann der Geradeaus-Fahrt. Zur Konstruktion finden sich in den Constructopedia lediglich Teilhinweise zum Zusammenbau eines Differentials¹ und einer Zahnstangenlenkung², so dass den Konstrukteuren und Programmierern zahlreiche kreative Freiräume zur Verfügung stehen.

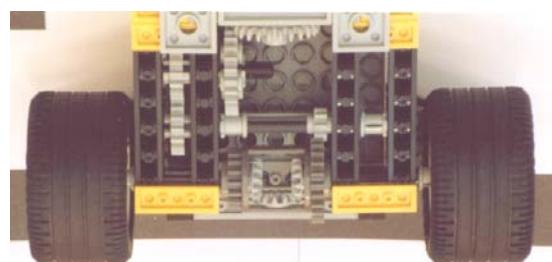
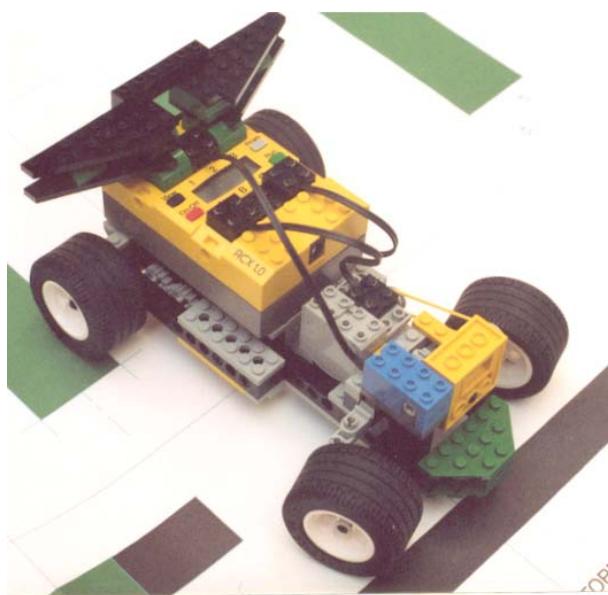
LEGO® MINDSTORMS™
Robotics Invention System™ 2.0 Test Pad

Abbildung oben: Differential Hinterachse

Abbildung links: Formel 1 Fahrzeug mit

- Zahnstangenlenkung
- Rotationssensor

¹ Lego-MINDSTORMS Constructopedia RIS 2.0, S. 102² Lego-MINDSTORMS Exploration Mars Constructopedia, S. 45

7. SCHLUSSBEMERKUNGEN

Am Ende einer Unterrichtsreihe sollte immer hinterfragt werden, inwieweit die didaktischen Überlegungen umgesetzt werden konnten und die damit verbundenen Lernziele als erreicht angesehen werden können. Weiterhin sollte über die Methodenauswahl reflektiert werden. Aufgrund der Rahmenbedingungen an der Musterschule im Schuljahr 2002/2003 – Lieferengpässe und Stundenausfall am Schuljahresende – können sich meine Erfahrungen nur auf die ersten Stunden der gesamten Unterrichtsreihe beziehen. Da die weiteren Stunden aber einem in den Kapiteln Didaktik und Methodik beschriebenen Gesamtkonzept folgen, sollten auch diese Stunden zu Lernerfolgen bei den Schülerinnen und Schülern führen.

Überraschend war für mich die Faszination der Schülerinnen und Schüler, die von den Robotern ausging. Diese Motivation war während der gesamten Unterrichtszeit spürbar und trug wesentlich zum Erfolg der Unterrichtsreihe bei. Als Physiklehrer war ich überaus erfreut über die konstruktiven Roboterentwicklungen, die auf fundierte Kenntnisse und Fertigkeiten der Konstrukteure in den Teildisziplinen Mechanik und Elektrizitätslehre schließen lassen. Gleichzeitig waren die Schülerinnen und Schüler aber auch als Programmierer gefordert, indem sie Fähigkeiten aus dem Programmierkurs auf die Lego-Software übertragen mussten. Ganz im Sinn des Spiralcurriculums wurden hier Erkenntnisstrukturen gefestigt und allgemeine Problemlösestrategien erworben. Da es im Bereich Schule nur selten Möglichkeiten zum fachübergreifenden Unterricht gibt, sollte die Chance Lego-Roboter genutzt werden, um fachübergreifend physikalisch-technische und mathematisch-informatorische Aufgabenstellung zu behandeln. Am Beispiel des Dosenwettbewerbs kann der vollständige Entwicklungszyklus einer Software nachvollzogen werden. Objektorientierte Elemente lassen sich propädeutisch einbinden und können im Informatikunterricht der Sekundarstufe II vertieft werden.

Trotz dieser anspruchvollen Aufgaben ist ein naiver Zugang realisierbar. Er ist auch von Gruppen ohne Vorkenntnisse in Informatik leistbar. Lediglich im Bereich der Programmierung muss dann der Übungszeitrahmen erweitert werden. Eine separate Projektwoche unter dem Thema Lego-MindStorms halte ich daher auch für grundsätzlich durchführbar und empfehlenswert.

Ein wesentlicher Faktor zum Erfolg dieser Arbeit liegt meiner Meinung nach auch in der Methode „Gruppenarbeit“ begründet. Im Gegensatz zum „Normalunterricht“ waren die Schülerinnen und Schüler wesentlich aktiver am Unterrichtsgeschehen beteiligt. Leistungsschwächere Schülerinnen und Schüler profitierten von der Gruppenarbeit, in die sie sich leichter einbringen konnten als bei fragend-entwickelnden Unterrichtsformen, die häufig von Experten dominiert werden. Zur Integration der Schülerinnen und Schüler trug auch die Übertragung von Spezialaufgaben (Teammanager, Material-, Kommunikations- und Informationsspezialist) bei, da diese unabhängig vom Leistungsniveau im Fach Informatik ausgeführt werden können. In den verschiedenen Gruppen konnte ein positives „Wir“-Gefühl und eine deutliche Identifikation mit dem eigenen Roboter festgestellt werden.

Die Unterrichtsreihe Lego Mindstorms war daher ein erfolgreicher Abschluss der WPU-Kurse, zu deren Durchführung ich nur anregen kann.

8. LITERATURVERZEICHNIS

BAUMANN R.: Didaktik der Informatik
Klett, Stuttgart (1996)

BOUZEGHOUB M., GARDARIN G. & VALDURIEZ P.: Object Technology
International Thomson Computer Press, Boston (1997)

BUSSMANN H. & HEYMANN H.-W.: Computer und Allgemeinbildung
Neue Sammlung 1 (1987)

DENERT E.: Software Engineering
Springer, Berlin (1991)

GUDJONS H.: Handlungsorientierung als methodisches Prinzip im Unterricht
In **TERHART E.**: Lehr-Lern-Methoden, Weinheim (1997)

GUDJONS H.: Projektunterricht begründen –
Sozialisationstheoretische und lernpsychologische Argumente
In **BASTIAN J. & GUDJONS H.**: Das Projektbuch II,
Bergmann & Helbig, Hamburg (1990)

GREHAM R., MOOTE R. & CYLIAX I.: Real Time Programming,
Addison-Wesley, Reading, Mass. (USA) (1998)

HESSISCHES KULTUSMINISTERIUM: Lehrplan Informatik, Gymnasialer Bildungsgang,
Jahrgangsstufe 11 bis 13, Wiesbaden (2003)

HUBWIESER P.: Didaktik der Informatik
Springer Verlag, Berlin (2001)

LEGO MINDSTORMS: Exploration Mars Constructopedia
The Lego Company (2000)

LEGO MINDSTORMS: Robotics Invention System 1.5 Constructopedia
The Lego Company (1999)

LEGO MINDSTORMS: Robotics Invention System 1.5 User Guide
The Lego Company (1999)

LEGO MINDSTORMS: Robotics Invention System 2.0 Constructopedia
The Lego Company (2000)

LEGO MINDSTORMS FÜR SCHULEN: Lösen einer Teamaufgabe mit dem Lego Set #9793/9794
Lego educational division Vertrieb: Technik LPE GmbH, Eberbach (2000)

LEHMANN E.: Komplexe Systeme – Teil 2: Eine fundamentale Idee im Informatikunterricht
In Log In 15 (1995) H. 1 S. 29 - 37

Louis D. & Müller P.: Roboter programmieren
Markt + Technik Verlag, München (2001)

Modrow E.: Virtuelle Lehrerweiterbildung Informatik in Niedersachsen, Didaktik Teil1
www.eikemeyershof2.de/datenVLIN/DateienKurs1/weiteres/didakt1.pdf
CD: Literatur/Modrow.pdf

Połoczek J.: Lego MindStorms (2002)
www.bildung.hessen.de/abereich/inform/skii/material/delphi/lego/index.htm

RoboLAB: Erfinder Set 9790 – Arbeitshandbuch
Lego dacta Vertrieb: Technik LPE GmbH, Eberbach (1998)

RoboLAB: Erste Schritte – Lehrerhandbuch zur RoboLab-Software
Lego dacta Vertrieb: Technik LPE GmbH, Eberbach (1998)

Röhner G.: Didaktik der Informatik – Projektarbeit
Handreichung zur 5. WBK VII Fortbildung, Kassel (2002)

Schwil A.: Vorlesung zur Didaktik , Wintersemester 1996/1997
www.-cg-hci.informatik.uni-oldenburg.de/~cgserg/Didaktik99/VLSchwil_Did-d-Inf.pdf
CD: Literatur/Schwil96.pdf

BEZUGSADRESSEN

LPE Technische Medien GmbH, Postfach 1121, 69401 Eberbach
Tel.: 06271 / 9234-10, Fax: 06271 / 9234-20,
E-Mail: Info@technik-lpe.com, Internet: www.technik-lpe.com

ANHANG

- ARBEITSBLÄTTER
- ANHÄNGE ZU ARBEITSBLÄTTERN
- PROGRAMM-CODES

ARBEITSBLATT 1**FÜR ARBEITSGRUPPE:**

(Arbeitsgruppenname)

Mitglieder der Arbeitsgruppe: _____

Bearbeitet die nachstehenden Aufgaben und führt rechts Protokoll.
 Der Kommunikationsspezialist sammelt die Protokolle.

Aufgaben**Erledigt**

1. Verbindet den IR-Tower mit dem PC
 USB-IR-Tower verfügen über ein festes Anschlusskabel.
 IR-Tower für die serielle Schnittstelle werden über ein 9-poliges
 Kabel mit der COM1-Schnittstelle am PC verbunden.
 Diese Tower benötigen eine 9 V Batterie.



1

2. Startet das Lego-Programm auf dem
 Desktop im Ordner „INFORMATIK“
 Wählt die Option „AUSFÜHREN“ (STARTSYMBOL)
 Meldet euch unter eurem Arbeitsgruppennamen an
 Startet die „TOUR“
 Wählt die Option „LEKTIONEN“ und beginnt mit den
 „TRAININGSLEKTIONEN“ für Anfänger und bearbeitet
 die folgenden Optionen:
- „EINFÜHRUNG“
 - „1. DAS IST EIN RCX“
 - „2.BAUEN DES ROBOTERS (ROVERBOT)“
 (Bauanleitung Constructopedia Seite 10 – 17)
 - „3.PROGRAMMIERGRUNDLAGEN“

1

1

1

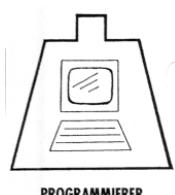
1

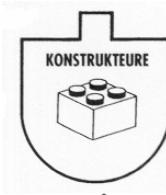
1

1

1

3. Legt eure Rollen für die nächste Teamsitzung fest (siehe Anlage 1)
- Jede/r ist Konstrukteur **oder** Programmierer
 - Wählt **eine** Zusatzqualifikation













4. Falls noch etwas Zeit verbleibt, könnt ihr euren Roverbot beschreiben und dekorieren. Beispiele findet ihr u.a. in der Constructopedia auf den Seiten 88 und 89. Seid kreativ!

ANLAGE ZUM ARBEITSBLATT 1:**Aufgaben der Spezialisten****Teammanager:**

- Behält die Aktivität des Teams im Auge
- Stellt sicher, dass das Projekt in Übereinstimmung mit dem Zeitplan durchgeführt wird
- Leitet die Gruppengespräche
- Wenn Unstimmigkeiten auftreten, sorgt er dafür, dass sich das Team über die weitere Vorgehensweise einigt

**Materialspezialist:**

- Kontrolliert die Verwendung der Hardware, Software, Bauelemente und anderer Materialien
- Verteilt Material und bestellt oder beschafft zusätzliches Material, falls benötigt
- Sorgt dafür, dass jeder das Material ordnungsgemäß aufbewahrt

**Informationsspezialist:**

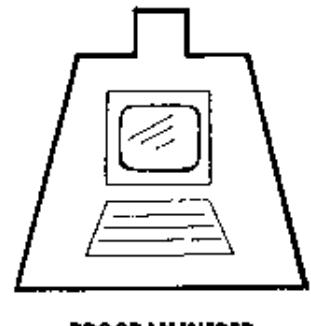
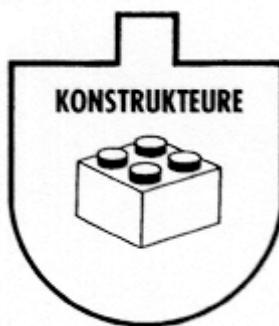
- Stellt alle Informationen für das Projekt zusammen und stellt sie den anderen Teammitglieder zur Verfügung
- Recherchiert im Internet
- Plant und organisiert das arbeitsteilige Vorgehen

**[Kommunikationsspezialist:]**

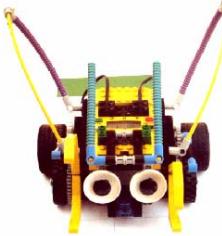
(nur bei 4-er Gruppen, in 3-er Gruppen übernimmt der Informations-spezialist diese Aufgaben zusätzlich)

- Organisiert mit den anderen Kommunikationsspezialisten den Wettbewerb am Projektende
- Repräsentiert die Gruppe nach aussen
- Bereitet die Projektberichte vor und führt die Protokolle

Zusätzlich zu diesen Verpflichtungen nehmen alle Schülerinnen und Schüler Aufgaben als Konstrukteure oder Programmierer wahr. Bei diesen Tätigkeiten sollt ihr euch im Team abwechseln. Die Einführungslektionen sollt ihr gemeinsam durcharbeiten.

**PROGRAMMIERER**

ANLAGE ZUM ARBEITSBLATT 1:**Fahrzeugbeschreibung**

Mobiler_Roboter	
Attribute	
-typ:	Roverbot
-eigenname:	_____
-anzahlMotoren:	1: () 2: () 3: ()
-antrieb:	Räder () Ketten ()
-sensoren:	Licht () Druck () Rotation ()
-dekoration	_____
Methoden	
+vorwärts()	_____

ARBEITSBLATT 2**FÜR ARBEITSGRUPPE:** _____

Teammanager: _____

Informationsspezialist: _____

Materialspezialist : _____

Kommunikationsspezialist: _____

Aufgaben**Erledigt**

1. Die Spezialisten besorgen:
 - Materialspezialist: Gruppenbox, geladene Akkus, IR Tower
 - Kommunikationsspezialist: PC
 - Informationsspezialist: Arbeitsblatt, Protokollvorlage
 - Teammanager: Allgemeiner Aufbau der Baukästen

2. Bearbeitet die restlichen Trainingslektionen für Anfänger

In eurem Protokoll der letzten Stunde steht wie weit ihr gekommen seit.
„HAUPTMENÜ“ → „LEKTIONEN“ → „TRAININGSLEKTIONEN“

 - „3. PROGRAMMIERGRUNDLAGEN“ (falls noch nicht erledigt)
 - „4. WEITERE BEISPIELE“
 - „5. SENSOREN“
 - „6. PROGRAMMABLAUF“



3. Baut eine Fernbedienung für euren Roverbot. Eine Bauskizze findet ihr links auf der Seite. Benutzt zwei ganz lange Kabel zur Verbindung der Fernbedienung mit den Eingängen 1 und 3 am RCX.

Abbildung: Fernbedienung
mit zwei Drucksensoren

Aufgaben**Erledigt**

4. Startet über das „HAUPTMENÜ“ → „PROGRAMM“ → „ROBOTER AUSWÄHLEN“ → „ROVERBOT“ → „DIESEN ROBOTER PROGRAMMIEREN“ → „DATEI“ → „ÖFFNEN“

und öffnet das Programm: S_Sensor.lsc (K:\TOOLS\WPU10\LEGO\S_SENSOR.LSC)
Beschreibt das Fahrverhalten:



und öffnet das Programm G_Sensor.lsc (K:\TOOLS\WPU10\LEGO\G_SENSOR.LSC)
Beschreibt das Fahrverhalten:



Warum ist die 2. Programmvariante besser?



5. Beachtet im Programm

- das Kommentarfeld

Das Kommentarfeld erhält man über „BEARBEITEN“ → „KOMMENTAR“

- das Programmende-Zeichen

Das Programmende-Zeichen erhält man über
„KLEINE BLÖCKE“ → „ERWEITERT“ (Letzter Eintrag)

- das Logbuch beim Speichern der Programme

6. Schreibt ein Programm für die Fernbedienung:

Keine Taste gedrückt: Geradeaus Fahrt (0,1 s)

Linke Taste gedrückt: Linkskurve (0,1 s)

Rechte Taste gedrückt: Rechtskurve (0,1 s)

Beide Tasten gedrückt: Rückwärts Fahrt (0,1 s)

Gesamtdauer 30 Sekunden



Tipps: • Erstellt ein Zustandsdiagramm für die Motoren (Anlage)

- Warte-bis-Schleife mit Timer verwenden

- Ja/Nein-Abfragen schachteln!

- Übertragen und Abspeichern

- Kommentar und Logbucheintrag nicht vergessen

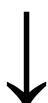
Warum versagt die von LEGO vorgeschlagene Sensorprogrammierung in diesem Fall?

ANLAGE ZUM ARBEITSBLATT 2:**Zustandsdiagramme**

Ereignis: Sensor 1: **0** ; Sensor 2: **0**

Zustandsname: „GeradeAus“

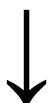
	Ein/Aus oder Leistungsstufe 0..8	Drehrichtung vorwärts, rückwärts
Motor A	E	V
Motor C	E	V



Ereignis: Sensor 1: ____ ; Sensor 2: ____

Zustandsname:

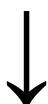
	Ein/Aus oder Leistungsstufe 0..8	Drehrichtung vorwärts, rückwärts
Motor A		
Motor C		



Ereignis: Sensor 1: ____ ; Sensor 2: ____

Zustandsname:

	Ein/Aus oder Leistungsstufe 0..8	Drehrichtung vorwärts, rückwärts
Motor A		
Motor C		



Ereignis: Sensor 1: ____ ; Sensor 2: ____

Zustandsname:

	Ein/Aus oder Leistungsstufe 0..8	Drehrichtung vorwärts, rückwärts
Motor A		
Motor C		

ANLAGE ZUM ARBEITSBLATT 2**FÜR LEHRERINNEN UND LEHRER**

Programmcode: S_SENSOR.lsc



Programmcode: G_SENSOR.lsc

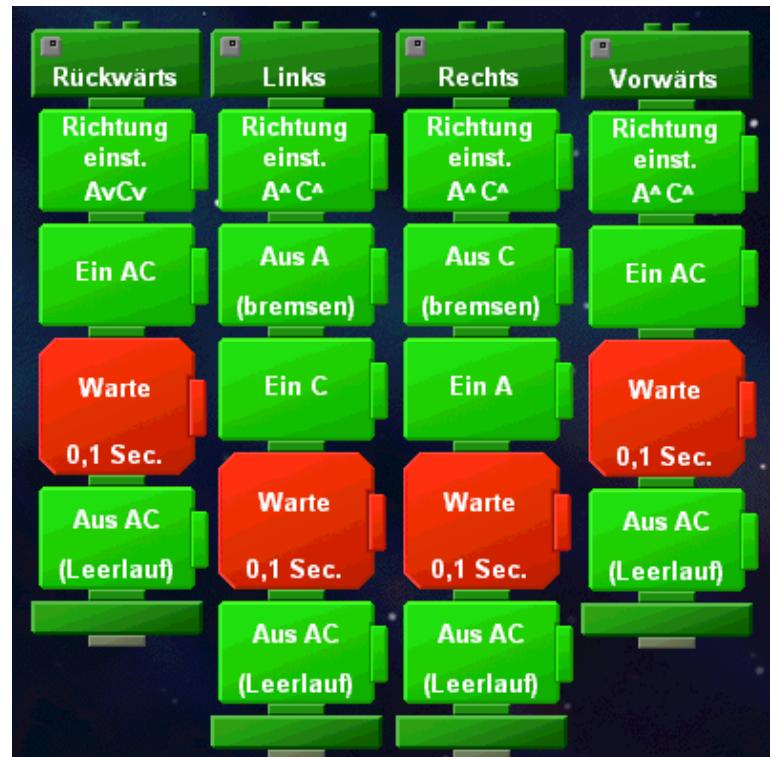


ANLAGE ZUM ARBEITSBLATT 2**FÜR LEHRERINNEN UND LEHRER**

Programmcode: FERNBEDIENUNG.LSC



Abbildungen:

Hauptprogramm (oben)
Methoden (rechts)

ARBEITSBLATT 3**FÜR ARBEITSGRUPPE:** _____**Bitte heute die Spezialistenfunktionen tauschen!**

Teammanager: _____

Informationsspezialist: _____

Materialspezialist : _____

Kommunikationsspezialist: _____

Aufgaben**Erledigt**

1. Die Spezialisten erledigen die Startvorkehrungen
(Ggf. hilft ein Blick in das letzte Protokoll)
2. Bearbeitet einige Aufgaben „Challenges“ unter
„HAUPTMENÜ“ → „LEKTIONEN“ → „AUFGABEN“
 - „FILMINTRO“ anschauen
 - „ROVERBOT“ → „AUFGABE EINGEBEN“
 - Stufe 1, Schwierigkeitsgrad: einfach
Auswahl: „GUARDBOT“, „DANCER“ oder „PUSHBOT“
(Einzelstoßfänger, Bauanleitung Constructopedia S. 26 - 29)
 - Stufe 2, Schwierigkeitsgrad: mittel
Auswahl: „EXPLORER“, „LIGHT SEEKER“ oder "WALL HUGGER"
(Doppelstoßfänger, Bauanleitung Constructopedia S. 30 - 33)

Beim „LIGHT SEEKER“
GROßen BLOCK „LICHT SUCHEN“
verwenden



- Stufe 3 Schwierigkeitsgrad: schwer
Auswahl: „LINE FOLLOWER“ oder “CLEAN SWEEP”
(Lichtsensor, Bauanleitung Constructopedia S. 34 - 35)

Ihr müsst nicht alle genannten Aufgaben aus dem ROVERBOT-Bereich bearbeiten. Es ist besser, eine geringere Anzahl von Aufgaben sorgfältig und vollständig zu bearbeiten, als viele Challenges nur teilweise zu erledigen. Die Programmbeispiele sind unvollständig und müssen von euch ergänzt werden.

Für die Aufgaben „BOOMERANG“, „MOZART“, „LIGHT DRIVE“ und „SOUND MAKER“ werden Variable benötigt. Wie diese bei Lego MindStorms deklariert und verwendet werden, erfahrt ihr im nächsten Übungsabschnitt für Fortgeschrittene.

Erledigt

3. Bearbeitet die restlichen Trainingslektionen für Fortgeschrittene
„HAUPTMENÜ“ → „LEKTIONEN“ → „TRAININGSLEKTIONEN“
 - 7. „KLEINE BLÖCKE“
 - 8. „EIGENE BLÖCKE“
 - 9. „VARIABLEN“
 4. Wählt im „HAUPTMENÜ“
die Optionen „PROGRAMM“ → „ROBOTER AUSWÄHLEN“ → “ROVERBOT“,
um in den Programmeditor zu gelangen

Programmiert einen Eigenen Block „DREHENLINKS“.
 - Der ROVERBOT soll sich für 1 Sekunde auf der Stelle drehen.
- Tipp: **Erstellt ein Zustandsdiagramm für die Motoren**
Beide Motoren entgegen gesetzt laufen lassen.
Verwendet den Großen Block „VORWÄRTS“ als Vorlage

 - Der ROVERBOT soll sich genau einmal um seine Achse drehen
 - Programmiert einen Eigenen Block „DREHENRECHTS“
 5. Die Trainingslektionen habt ihr jetzt erfolgreich abgeschlossen. Nun beginnt der kreative Teil der Arbeit. Für den Rest der Stunde können die **Konstrukteure** den Roverbot mit verschiedenen Antrieben versehen. Es gibt:
 - Kettenantriebe
 - „Beine“
 - diverse Radsätze
 - Elektrische Lampen
 - Dekorationsmaterialien (Klaue, Greifarme, Flügel,...)

Die **Programmierer** können

 - die Sensorabfragen modifizieren
 - Elektrische Lampen ansteuern
 - Musikbegleitungen der Aktionen programmieren
(Beispielprogramme findet ihr in „GROSSE BLÖCKE“ → „GLOBAL“)

Gebt euren ROVERBOT-Varianten Namen und beschreibt ihre Eigenschaften (Attribute) und Fähigkeiten (Methoden). Scannt ggf. ein Bild ein.

Verwendet die beiliegende Anlage zur Fahrzeugbeschreibung und heftet sie mit dem Arbeitsblatt als Protokoll ab.

ANLAGE ZUM ARBEITSBLATT 3:**Fahrzeugbeschreibung**

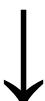
Mobiler_Roboter	Bild einscannen
Attribute	
-typ:	Roverbot
-eigenname:	_____
-anzahlMotoren:	1: () 2: () 3: ()
-anzahlLampen:	1: () 2: () 3: ()
-antrieb:	Räder () Ketten ()
-sensoren:	Licht () Druck () Rotation ()
-sound:	_____
-dekoration	_____
Methoden	

ANLAGE ZUM ARBEITSBLATT 3:**Zustandsdiagramme**

Ereignis:

Zustandsname: „DrehenLinks“

	Ein/Aus oder Leistungsstufe 0..8	Drehrichtung vorwärts, rückwärts
Motor A		
Motor C		



Ereignis:

Zustandsname: „DrehenRechts“

	Ein/Aus oder Leistungsstufe 0..8	Drehrichtung vorwärts, rückwärts
Motor A		
Motor C		

ANLAGE ZUM ARBEITSBLATT 3 FÜR LEHRERINNEN UND LEHRER

Übersicht der geänderten Programmcodes:

Stufe 1:



“GUARDBOT”



„DANCER”



“BOOMERANG”



“PUSHBOT”

ANLAGE ZUM ARBEITSBLATT 3 FÜR LEHRERINNEN UND LEHRER

Übersicht der geänderten Programmcodes:

Stufe 2:



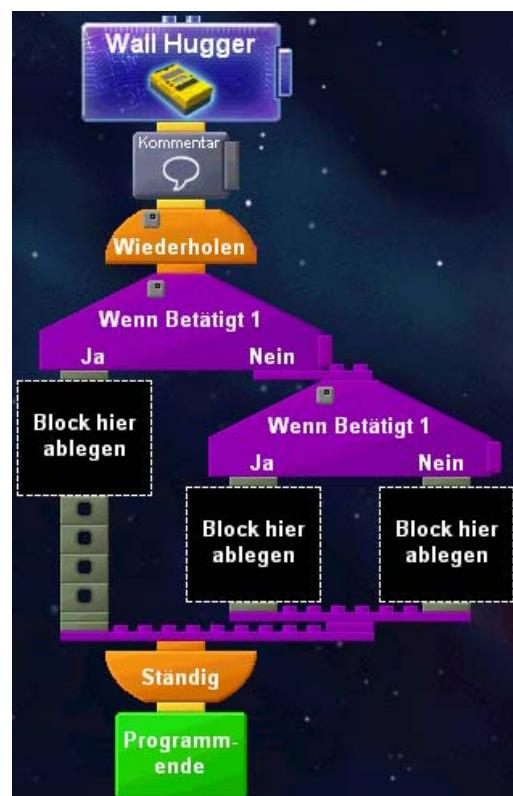
“EXPLORER”



“MOZART”



“LIGHT SEEKER”



“WALL HUGGER”

ANLAGE ZUM ARBEITSBLATT 3 FÜR LEHRERINNEN UND LEHRER

Übersicht der geänderten Programmcodes:

Stufe 3:



“LINE FOLLOWER”



“CLEAN SWEEP”



“SOUND MAKER”



“LIGHT DRIVE”

ANLAGE ZUM ARBEITSBLATT 3 FÜR LEHRERINNEN UND LEHRER

Programmcode der „EIGENEN BLÖCKE“:



PROGRAMMCODE „EIGENER BLOCK“:
„DREHENLINKS“



PROGRAMMCODE „EIGENER BLOCK“:
„DREHENRECHTS“

ARBEITSBLATT 4 FÜR ARBEITSGRUPPE: _____**Der Wettbewerb**

Damit habt ihr alle Einführungslektionen erfolgreich absolviert. In den nächsten Stunden sollt ihr euch einer neuen Herausforderung („Challenge“) stellen. Der „Dosenwettstreit“ stellt eine Trainingsmission für Robotersysteme dar. Die Wettkampfregeln werden auf den nächsten Seiten beschrieben.

Überlegt euch mit welchem Fahrzeugtyp ihr in den Wettkampf ziehen wollt. In den verschiedenen Constructopédias (RIS 1.5, RIS 2.0, Mars-Explorer und Lego-MindStorms für Schulen) oder im Internet (Adressen siehe Anlage) findet ihr interessante Ideen, sowie wertvolle Hinweise und Konstruktionstipps. Es gelten folgende Vorschriften für die Fahrzeuge:

Außenmaße beim Start: Länge: max. 30 cm; Breite: max. 20 cm

Motoren: max. 3 Stück

Sensoren: max. 3 Stück

Zur Vorbereitung stehen euch die drei nächsten Doppelstunden zur Verfügung. Als Hilfe zur Planung findet ihr im Anhang ein Arbeitsraster für Konstruktion und Programmierung. Geht arbeitsteilig vor.

Die Trainingszeit auf der Originalwettkampffläche wird auf 2-mal 5 Minuten beschränkt. Die Trainingszeiten werden vom Teammanager beim Lehrer anmeldet. Die Kommunikationsspezialisten der Teams sollen sich treffen, um eventuell über alternative Wettkampfregeln zu beraten.

Doppel-Stunde	Funktionsbezeichnung			
	Teammanager	Materialspezialist	Informations-spezialist	Kommunikations-spezialist
1.				
2.				
3.				
Wett-kampf				

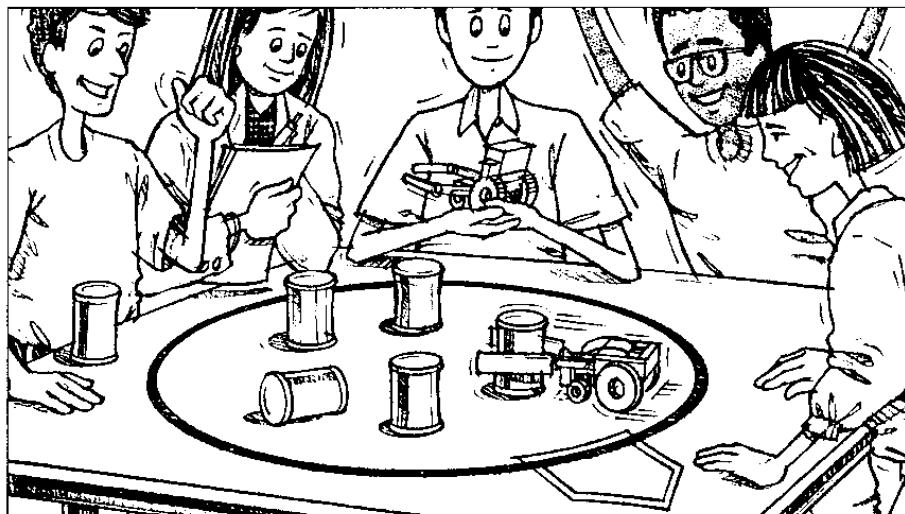
Die Funktionen rotieren jede Woche, so dass jedes Gruppenmitglied am Ende des Projekts einmal Teammanager, Material-, ggf. Kommunikations- und Informations-spezialist war.

ANLAGE ZUM ARBEITSBLATT 4:

Wettkampfregeln

DER DOSENWETTSTREIT 1/2¹

Das Dosenexperiment stellt die "Trainingsmission" für Robotersysteme dar. Sie wird auf den folgenden Seiten im Detail beschrieben. Verwenden Sie diese Aufgabe, falls Sie zum ersten Mal mit Ihrer Klasse eine Teamaufgabe bearbeiten. Die Teamaufgabe "Dosenexperiment" besteht darin, einen Roboter so zu bauen und zu programmieren, dass er innerhalb von 2 Minuten möglichst viele Dosen aus dem vorgegebenen Bereich entfernen kann.

Beschreibung**Wettkampf-Regeln**

Auf der nächsten Seite im Abschnitt "Alternative Regeln" finden Sie Vorschläge dazu, wie Sie den Schwierigkeitsgrad der Aufgabe abändern können.

- 1) Die Roboter befinden sich zu Beginn innerhalb der Start-Box in Richtung zur Kreismitte. Der Lichtsensor darf dabei über die Start-Box hinweg in die Kreisgrenze hineinreichen.
- 2) Die Roboter dürfen die Dosen aus dem Kreis schieben, tragen, kicken, schlagen, rollen oder werfen.
- 3) Eine Dose wird dann als "draußen" betrachtet, wenn sie vollständig außerhalb der Kreisgrenze liegt.
- 4) Das Berühren eines Roboters während einer Runde disqualifiziert diesen für den Rest der laufenden Runde. Die Dosen, die bis zu diesem Zeitpunkt bereits aus dem Kreis entfernt wurden, werden gezählt. Diese ergeben die Punktzahl, die der Roboter in dieser Runde erreicht hat.
- 5) Die Roboter dürfen die unmittelbar außerhalb des Kreises liegende Fläche lediglich zum Wenden benutzen.
- 6) Jede Runde dauert genau 2 Minuten. Nach Ablauf dieser Zeit muss der Roboter anhalten, und alle Dosen außerhalb der Kreisgrenze werden gezählt. Diese ergeben die Punktzahl für die abgelaufene Runde.
- 7) Bleibt ein Roboter über 30 Sekunden lang außerhalb der Kreisgrenze stehen, ist die Runde für ihn vorüber. Die Dosen, die bis zu diesem Zeitpunkt bereits aus dem Kreis entfernt wurden, werden gezählt. Diese ergeben die Punktzahl, die der Roboter in dieser Runde erreicht hat.
- 8) Je nach Absprache kann ein Team-Wettkampf 1-3 Runden umfassen, wobei zur Ermittlung des Siegers der Durchschnitt aus den Punktzahlen aller Runden berechnet wird.
- 9) Die funktionellen Bauteile aller Roboter (d.h. die Teile, die der Fortbewegung dienen) dürfen nur aus LEGO-Elementen bestehen, die dem 9793/94 Erfinder-Set II entnommen sind. Andere Materialien (z.B. Papier, Stoff) können jedoch zur Dekoration der Roboter verwendet werden.

¹ Quelle: Lego MindStorms for Schools, Seite 41; Lego Educational Division

DER DOSENWETTSTREIT 2/2¹

Alternative Regeln	
STUFE 1	Regeln zur Genauigkeit Es gelten dieselben Regeln wie zuvor beschrieben, allerdings werden nur die Dosen gezählt, die außerhalb des Kreises aufrecht stehen.
STUFE 2	Regeln zur Lichtwahrnehmung Es gelten dieselben Regeln wie zuvor beschrieben, allerdings muß sich der Lichtsensor zu Beginn innerhalb der Start-Box befinden. Sumo-Regeln Es gelten dieselben Regeln wie zuvor beschrieben, allerdings dürfen dieses Mal nur volle, ungeöffnete Dosen verwendet werden.
STUFE 3	Transportregeln Es gelten dieselben Regeln wie zuvor beschrieben, allerdings werden nur die Dosen gezählt, die aufgehoben und auf dem Roboter gelagert werden.
STUFE 4	Regeln zur Eingangssuche Es gelten dieselben Regeln wie zuvor beschrieben, allerdings müssen die Roboter zu Beginn in der Start-Box von der Kreisgrenze abgewendet aufgestellt werden. Jeder Roboter muss zunächst die Kreisgrenze finden, bevor er die Start-Box verlässt und die Dosen aus dem Kreis entfernt. Regeln zur Ausgangssuche Es gelten dieselben Regeln wie zuvor beschrieben, allerdings werden nur die Dosen gezählt, die durch die Start-Box hindurch den Kreis verlassen.

EINIGE NÜTZLICHE INTERNETADRESSEN

<http://www.legomindstorms.com/>

Eure Mitgliedsnummer lautet: AKGDKYE2ACJP
Die Seriennummer eines RCX 720995

<http://www.lego.com/dacta/home.asp>

<http://www.forwiss.uni-erlangen.de/MuT/bauanleitung.html>

http://www.informatik.fh-hamburg.de/~lego/Projekte/body_projekte_deutsch.html

<http://www.crynwrl.com/lego-robotics/>

<http://www.firstlegoleague.org/de>

http://www.dragonslair.ch/project/WebSite_Robotics/doku/Roboter_HW.pdf

Ihr könnt auch eine Suchmaschine verwenden!

¹ Quelle: Lego MindStorms for Schools, Seite 42; Lego Educational Division

ANLAGE ZUM ARBEITSBLATT 4: Spielfläche

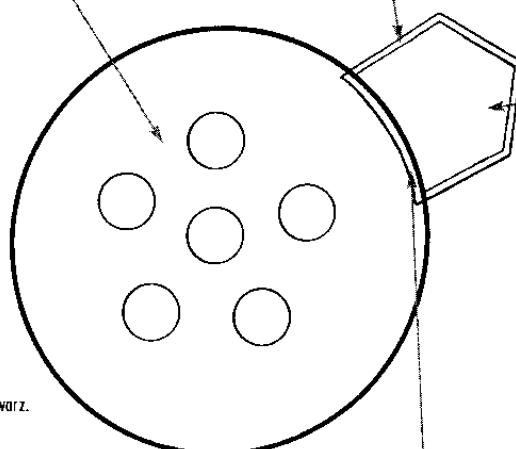
Die Spielfläche¹ kann gegebenenfalls im Unterricht erstellt werden.

Erstellt das Aktionsfeld für euer "Dosenexperiment entsprechend folgender Angaben.

Das Aktionsfeld für das Dosenexperiment

Es ist am besten, wenn ihr die Dosen leer und teilweise mit Sand oder sauberer Katzenstreu füllt. Versiegelt die Öffnungen mit Hilfe von Klebeband.

Fünf bis sieben 375 ml-Dosen werden innerhalb des Kreises ca. 23 Zentimeter vom Kreismittelpunkt entfernt aufgestellt.



Die Linien sollten 2 Zentimeter breit sein. (Falls ihr PVC-Elektroband verwendet, sollte es ebenfalls ca. 2 cm breit sein.)

Die Start-Box ist an der längsten bzw. breitesten Stelle 25 cm x 25 cm groß. Die Start-Box hat die Form eines Hauses, um die Alternativeufgabe auf Niveau 4, "Eingangssuche", interessanter zu gestalten. Die maximale Größe des Roboters beträgt 25 cm x 25 cm (passt genau in die Start-Box). Die Start-Box ist grün umrandet.

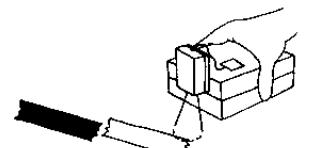
Die Kreislinie ist schwarz.

Der Teil der Kreislinie, der an der Start-Box anliegt, hat innerhalb der schwarzen Kreislinie einen grünen Streifen.

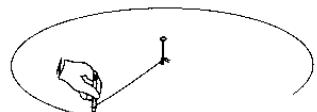
Wie wird das Aktionsfeld gebaut?

Falls ihr in eurem Raum einen Linoleum- oder Fliesenboden habt, so verwendet einfach diesen anstatt des Sperrholzes.

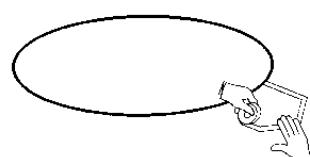
1) Sucht einen sauberen, glatten, hellen Tisch oder ein Stück glattes Laminat oder Sperrholz in der Größe von 122 cm x 244 cm aus.



2) Zeichnet mit Hilfe einer Schnur einen Kreis mit einem Radius von 45 cm (ergibt einen Kreisdurchmesser von 90 cm).



3) Malt oder klebt eine Kreisgrenze, indem ihr der gezeichneten Linie folgt. Falls ihr dazu Klebeband verwendet, müsst ihr daran denken, jegliche Falten zu glätten, um die Bewegungen des Roboters nicht zu behindern.



4) Messt und markiert die Start-Box. Malt die Umrandung grün oder verwendet grünes Klebeband.

5) Fahrt mit dem RCX über die Fläche und erfasst mit Hilfe der Ansichtstaste ("View") die Lichtwerte für weiße, grüne und schwarze Flächen.

6) Besorgt fünf bis sieben 375 ml-Dosen. Leert sie und füllt sie anschließend teilweise mit Sand oder Katzenstreu. Versiegelt die Öffnungen mit Klebeband.

¹ Quelle: Lego MindStorms for Schools, Seite 43; Lego Educational Division

Tätigkeitsliste 1 für Konstrukteure:

Name _____

Name _____

Erledigt / Aufgabennummer	Grundkonstruktion: Bauen und Testen		
	Unser Team hat:	Hinweise	Bemerkungen
1 K1.	sich für eine Grundkonstruktion entschieden.	Konstruktionsanleitungen für Basismodelle sind in den diversen Constructopedia zu finden.	
1 K2.	beschlossen, mit welcher Strategie zuerst probiert werden soll, die Dosen zu entfernen.		
Heute werden die Konstrukteure:			
1 K3.	die Pläne für eine ausgewählte Grundkonstruktion entwerfen.	Siehe oben, aber auch Eigenkonstruktionen sind erwünscht	
1 K4.	das Modell bauen und die Motoren hinzufügen.		
1 K5.	das Modell manuell auf seine Eigenschaften prüfen und die Konstruktion auf ihre Festigkeit testen.	RCX-Programm Nr. 1	
1 K6.	den Lichtsensor am Fahrzeug montieren.	Diskutiert die Lage des Sensors (vorn, hinten, links, Mitte, rechts).	
1 K7.	den Programmierern , den Roboter zeigen und erklären.	Punkt 6 abklären	
und falls noch Zeit übrig bleibt:			
1 K8.	die Grundkonstruktion verbessern und dekorieren.		

Tätigkeitsliste 2 für Konstrukteure:

Name _____

Name _____

Erledigt / Aufgabennummer	Fortbewegen und Wenden: Lasst Licht für Action sorgen !		
	Heute werden die Konstrukteure:	Hinweise	Bemerkungen
1 K1.	gemeinsam mit den Programmierern Motoren, Lichtsensor testen. Ggf. Abscanstrategie testen und verbessern.	Verwendet ein Testfeld, Sensoreingang 2 und RCX-Programm Nr.3 oder Nr. 5.	
1 K2.	mindestens 2 Rädervarianten oder Übersetzungsverhältnisse testen. Optimiert die Fahrgeschwindigkeit.	Siehe Infoblatt für Konstrukteure RCX-Programm Nr.1 oder Nr. 5	
1 K3.	die Antriebsvarianten auf ihre Wendefähigkeit prüfen. Polt dazu einen Motoranschluss um (bei Programm Nr 1).		
1 K4.	gemeinsam mit den Programmierern das Fahren, Wenden und Abscannen der Fläche mit dem entwickelten Steuerprogramm testen.	Programm auf RCX-Speicherplatz Nr. 5 übertragen.	
und falls noch Zeit übrig bleibt:			
1 K5.	die Funktionalität des Fahrzeugs optimieren.	Werden alle Bereiche abgefahren ?	
1 K6.	ggf. eine Trainingsrunde auf Originalwettkampfarena an durchführen.	Achtung, es darf nur zweimal getestet werden!	

Tätigkeitsliste 3 für Konstrukteure:

Name _____

Name _____

Erledigt / Aufgabennummer	Dosen sammeln mit Gefühl: Ganz bis zur Linie !		
	Heute werden die Konstrukteure:	Hinweise	Bemerkungen
1 K1.	gemeinsam mit den Programmierern eine Mechanik zum Entfernen der Dosen diskutieren.		
1 K2.	eine Mechanik zum Entfernen der Dosen entwickeln.	Siehe Infoblatt für Konstrukteure oder schaut in der Constructopedia nach.	
1 K3.	den Programmierern, den und erklären.		
1 K4.	die Wartezeit zum Verbessern der Funktionalität verwenden.	Massen gleichmäßig auf alle Achsen verteilen.	
1 K5.	gemeinsam mit den Programmierern den Mechanismus testen und optimieren.		
1 K6.	eine Trainingsrunde auf der Originalwettkampfarena anmelden und durchführen.		
	und falls noch Zeit übrig bleibt:		
1 K7.	das Fahrzeug dekorieren.		

Tätigkeitsliste 4 für Konstrukteure:

Name _____

Name _____

Erledigt / Aufgabennummer	Am Tag X: Heute geht's ums Ganze !		
	Heute werden die Konstrukteure:	Hinweise	Bemerkungen
1 K1.	gemeinsam mit den Programmierern das Wettkampfprogramm mit Übungsdosens testen.		
1 K2.	die Dekoration vervollständigen.	Größe beachten: L: max. 30 cm B: max. 20 cm	
1 K3.	gemeinsam mit den Programmierern entscheiden, welche besonderen Tätigkeiten der Roboter sonst noch ausführen soll und wann er diese ausführen soll.	Es gibt eine B-Note für den künstlerischen Eindruck.	
1 K4.	eine Trainingsrunde auf der Originalwettkampfarena anmelden und durchführen.		
1 K5.	Batterien überprüfen.		
1 K6.	Roboter zum Wettkampf anmelden.	Viel Erfolg !	

Tätigkeitsliste 1 für Programmierer

Name _____

Name _____

Erledigt / Aufgabennummer	Erste Schritte mit dem RCX: Schleifen erzeugen und Programme laden		
	Unser Team hat:	Hinweise	Bemerkungen
1 P1.	sich für eine Grundkonstruktion entschieden.	Konstruktionsanleitungen für Basismodelle sind in den diversen Constructopedias zu finden.	
1 P2.	beschlossen, mit welcher Strategie zuerst probiert werden soll, die Dosen zu entfernen.		
	Heute werden die Programmierer:		
1 P3.	die Kommunikation zwischen IR-Tower und RCX herstellen.	Ggf. die Firmware downloaden	
1 P4.	ein Programm für eine Vorwärtsfahrt bis zur Kreislinie schreiben.	Lichtsensor am Eingang 2 anschließen. Programmplatz Nr. 5 verwenden.	
	Über den Teammanager Stand der Konstruktion (Punkt K7) abfragen. Ggf. Punkte P2 – P5 der Tätigkeitsliste 2 für Programmierer vorziehen.		
1 P5.	gemeinsam mit den Konstrukteuren das Programm am Fahrzeug testen.	Diskutiert die Lage des Sensors (vorn, hinten, links, Mitte, rechts).	
1 P6.	eine Strategie zum Abscannen der Fläche entwickeln und programmieren.		
	und falls noch Zeit übrig bleibt:		
1 P7.	mit Tätigkeitsliste 2 für Programmierer beginnen.		

Tätigkeitsliste 2 für Programmierer

Name _____

Name _____

Erledigt / Aufgabennummer	Seht ihr die Linie? Schaltet den Motor ein und hört die Melodie !		
	Heute werden die Programmierer:	Hinweise	Bemerkungen
1 P1.	gemeinsam mit den Konstrukteuren Motoren, Lichtsensor testen. Ggf. Abscanstrategie testen und verbessern.	Verwendet ein Testfeld, Sensoreingang 2 und RCX-Programm Nr.3 oder Nr. 5 Tipp: Zufallsvariable verwenden	
1 P2.	eine Aktionsmelodie programmieren und auf Programmplatz 4 speichern.	Tonfrequenzen siehe Infoblatt 2 für Programmierer	
1 P3.	ein Programm schreiben, das abspielt, wenn der Lichtsensor die Kreislinie erreicht.	Programm auf Speicherplatz 4 downloaden.	
1 P4.	den Konstrukteuren die Melodie vorspielen.		
1 P5.	Programme 4 und 5 kombinieren.		
	und falls noch Zeit übrig bleibt:		
1 P6.	eine spezielle Startmelodie komponieren.	Tonfrequenzen siehe Infoblatt 2 für Programmierer	

Tätigkeitsliste 3 für Programmierer

Name _____

Name _____

Erledigt / Aufgabennummer	Berühren und Loslegen: Mit Gefühl !		
	Heute werden die Programmierer:	Hinweise	Bemerkungen
1 P1.	gemeinsam mit den Konstrukteuren eine Mechanik zum Entfernen der Dosen diskutieren.		
1 P2.	eine Licht- oder Musikaktion programmieren, die beim Berühren der Dose startet.	Achtung, wird ein 3. Motor verwendet, kann keine Lampe angeschlossen werden	
1 P3.	den Konstrukteuren die ne		
1 P4.	ein Programm zum Entfernen der Dosen entwickeln.		
1 P5.	gemeinsam mit den Programmierern den Mechanismus testen und optimieren.		
1 P6.	eine Trainingsrunde auf der Originalwettkampfarena anmelden und durchführen.		
	und falls noch Zeit übrig bleibt:		
1 P7.	Programmteile kommentieren.		

Tätigkeitsliste 4 für Programmierer:

Name _____

Name _____

Erledigt / Aufgabennummer	Am Tag X: Heute geht's ums Ganze !		
	Heute werden die Konstrukteure:	Hinweise	Bemerkungen
1 P1.	gemeinsam mit den Konstrukteuren das Wettkampfprogramm mit Übungsboxen testen.		
1 P2.	Aktionsmelodien für Start, Linien und Dosenberührung testen.		
1 P3.	gemeinsam mit den Konstrukteuren entscheiden, welche besonderen Tätigkeiten der Roboter sonst noch ausführen soll und wann er diese ausführen soll.	Es gibt eine B-Note für den künstlerischen Eindruck.	
1 P4.	eine Trainingsrunde auf der Originalwettkampfarena anmelden und durchführen.		
1 P5.	Batterien überprüfen.		
1 P6.	Roboter zum Wettkampf anmelden.	Viel Erfolg !	

Infoblatt 1 für Konstrukteure¹:

LEGO Educational Division Kopiervorlage

KONSTRUKTEUR
Informationsblatt 1 von 4

10

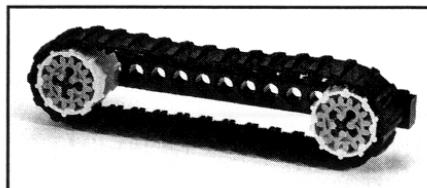
Name _____
Klasse _____
Datum _____



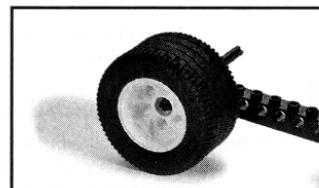
Wie kann ich erreichen, dass mein Roboter ...

Bodenhaftung bekommt?

Ketten

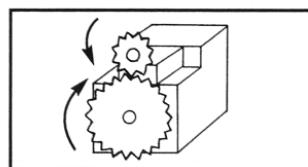


Breite Räder

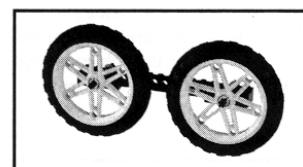


sich schneller fortbewegt?

Übersetzung

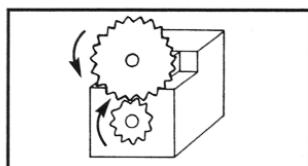


Große Räder

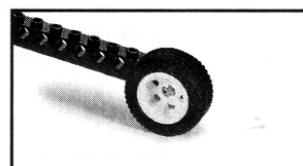


sich langsamer fortbewegt?

Untersetzung

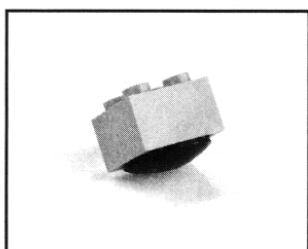


Kleinere Räder

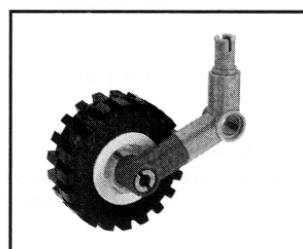


möglichst wendig ist?

Gleitkufe



Bewegliche Räder



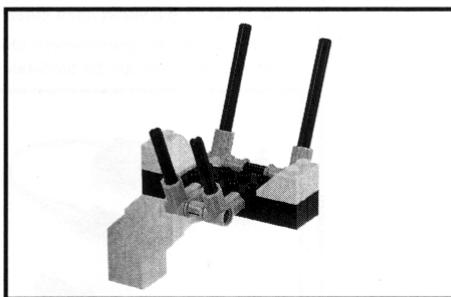
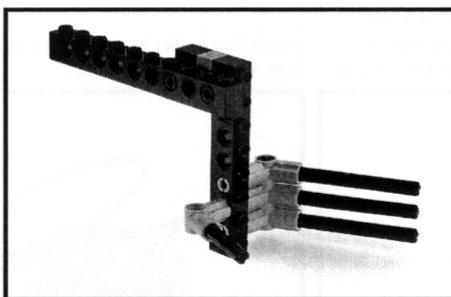
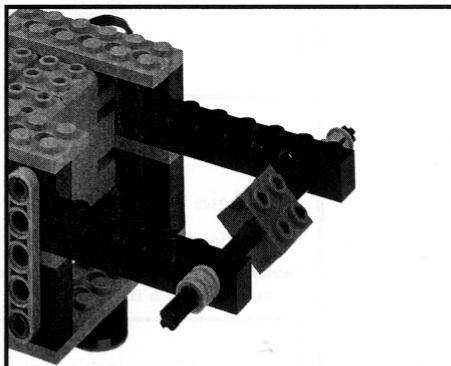
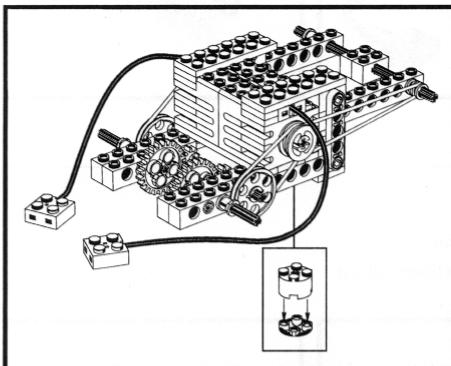
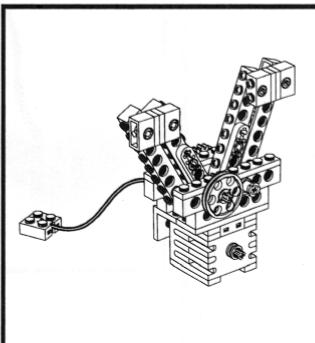
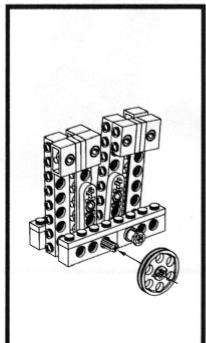
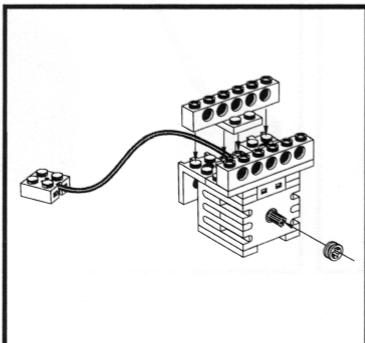
¹Quelle: Lego MindStorms for Schools, Seite 56; Lego Educational Division

Infoblatt 2 für Konstrukteure¹:

LEGO Educational Division Kopiervorlage

KONSTRUKTEUR
Informationsblatt 2 von 4

11

Name _____
Klasse _____
Datum _____**Wie kann ich erreichen, dass mein Roboter ...****die Dosen wegschiebt?****die Dosen wegkickt?****die Dosen greift?**¹Quelle: Lego MindStorms for Schools, Seite 57; Lego Educational Division

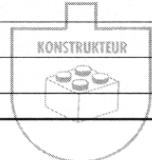
Infoblatt 3 für Konstrukteure¹:

LEGO Educational Division Kopiervorlage

KONSTRUKTEUR
Informationsblatt 3 von 4

12

Name _____
Klasse _____
Datum _____



Wie kann ich erreichen, dass mein Roboter ...

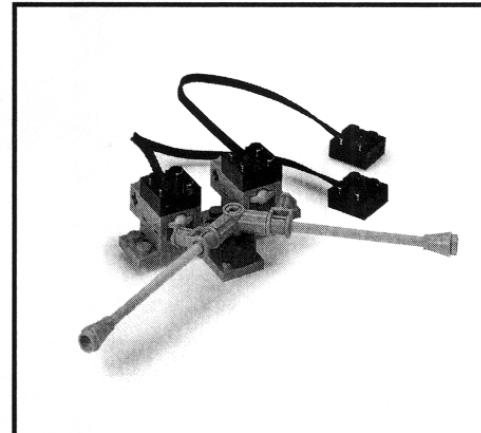
die Dosen fühlt?

Diese Stoßstangen verwenden einen Druckschalter.

Sprecht mit den Programmierern über ein Beispielprogramm, mit dem ihr die Stoßstangen testen könnt.



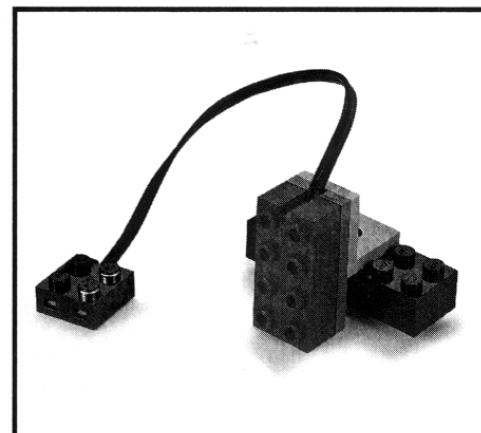
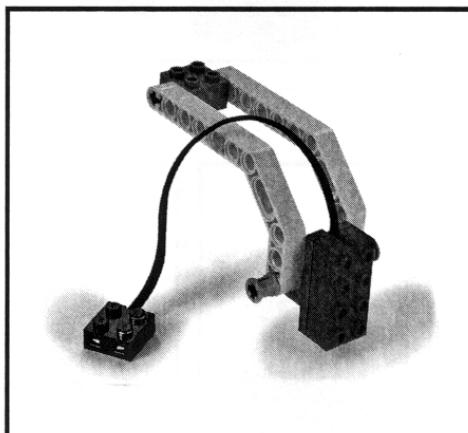
Dieses Paar Fühler kann anzeigen, welche Seite des Roboters die Dose berührt.



die Linie erkennt?

Schließt den Lichtsensor an Eingang 2 an.

Gestaltet mit schwarzem und grünem Klebeband auf weißem Papier ein Testfeld.



¹Quelle: Lego MindStorms for Schools, Seite 58; Lego Educational Division

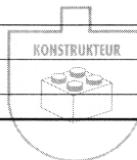
Infoblatt 4 für Konstrukteure¹:

LEGO Educational Division Kopiervorlage

KONSTRUKTEUR
Informationsblatt 4 von 4

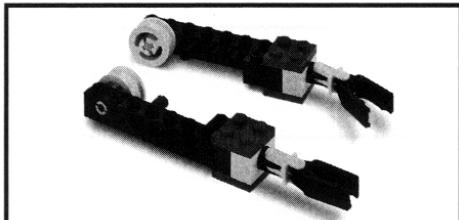
13

Name _____
Klasse _____
Datum _____

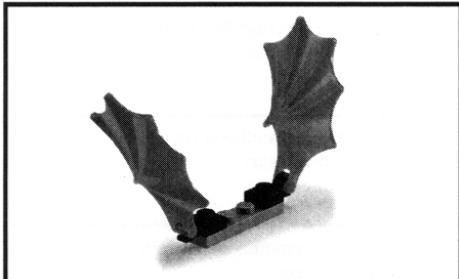


Wie kann ich das Aussehen meines Roboters verbessern durch Hinzufügen ...

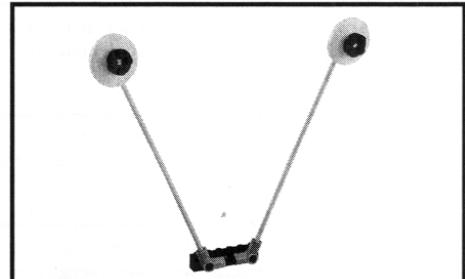
von Armen und Greifhänden?



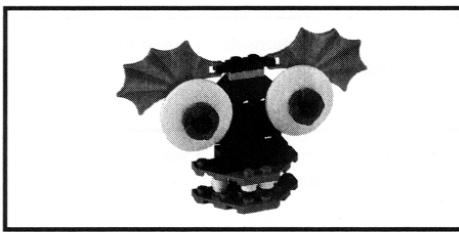
von Flügeln?



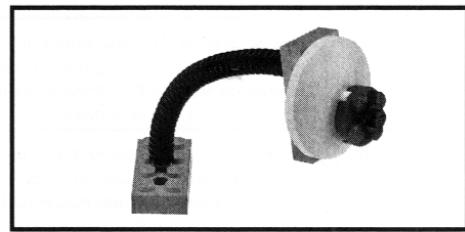
einer Antenne?



eines Gesichts?

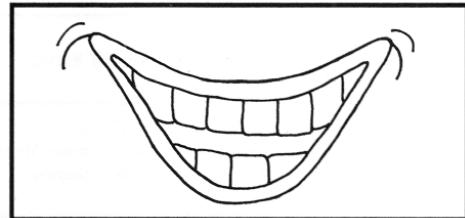


eines Auges?



Entwerft und zeichnet
eure eigenen
Dekorationen. Schneidet
sie aus und befestigt sie
an eurem Roboter.

von Dekorationen?



¹Quelle: Lego MindStorms for Schools, Seite 59; Lego Educational Division

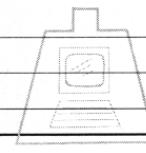
Infoblatt 1 für Programmierer¹:

LEGO Educational Division Kopiervorlage

PROGRAMMIERER Informationsblatt 1 von 6

18

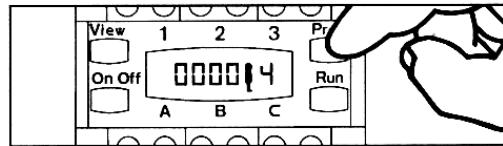
Name _____
Klasse _____
Datum _____



Wie kann ich erreichen, dass mein Roboter ...

... auf Programmplatz 4 heruntergeladen wird?

Drückt zunächst die RCX-Programmtaste (Prgm), bis eine 4 erscheint.

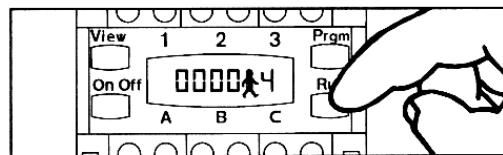


Klickt anschließend auf dem Bildschirm auf den Pfeil zum Herunterladen.

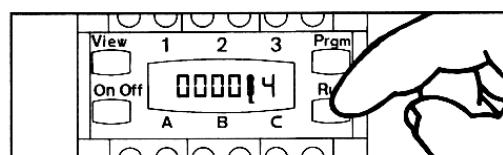


... das Programm startet und stoppt?

Drückt auf die RCX-Starttaste (Run), um das ausgewählte Programm auszuführen.

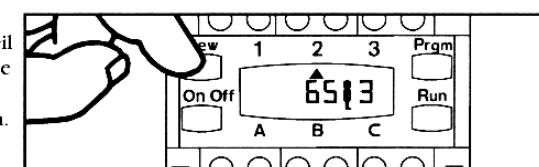


Drückt die Starttaste erneut, um das Programm anzuhalten.



... den Lichtwert anzeigt, den der Lichtsensor ermittelt?

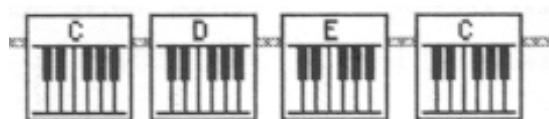
Schaltet den RCX ein. Drückt dann die Ansichtstaste (View), bis der Pfeil auf Eingang 2 deutet. Um Lichtwerte auf diese Weise anzeigen zu lassen, müsst ihr kein Programm ausführen.



¹Quelle: Lego MindStorms for Schools, Seite 64 und 65; Lego Educational Division

Infoblatt 2 für Programmierer:

Tonfrequenzen für Melodien



1.Oktave

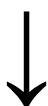
Ton	Frequenz [Hz]
C'	262
Cis'	277
D'	294
Dis'	311
E'	330
F'	349
Fis'	370
G'	392
Gis'	415
A'	440
Ais'	466
H'	494
C''	523

2. Oktave

Ton	Frequenz [Hz]
C''	524
Cis''	554
D''	587
Dis''	622
E''	659
F''	698
Fis''	740
G''	784
Gis''	831
A''	880
Ais''	932
H''	988
C'''	1047

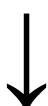
Beispiele für Melodien

- Hännschen klein (in C-Dur; Bespieldatei: HÄNSCHEN.LSC):
G' E' E' F' D' D' C' D' E' F' G' G' G' G' E' E' F' D' D' C' E' G' G' E'
- Happy Birthday (in C-Dur):
C' C' D' C' F' E' C' C' D' C' G' F' C' C' C'' A' F' E' D' C'' C'' A' F' G' F'
- Am besten programmiert ihr für jeden Ton einen Eigenen Block
Bsp.: Block "Ton_C_12", bedeutet C' für 0,4 s.
- Zwischen den Tönen eine Wartezeit (0,2 s) einbauen, damit sie einzeln wahrgenommen werden.
- Fertige Melodien sind unter „Große Blöcke“ → „Global“ → „Melodie“ zu finden.
Eigene Kreationen sind aber einzigartig.

ANLAGE ZUM ARBEITSBLATT 4: Kopiervorlage für Zustandsdiagramme

Ereignis(se):	_____
Sensoren, Timer	

Zustandsname:	_____	
Aktoren Motoren , Lampen & Sound	Ein/Aus, Bremsen oder Leistungsstufe 0..8	Drehrichtung vorwärts, rückwärts



Ereignis(se):	_____
Sensoren, Timer	

Zustandsname:	_____	
Aktoren Motoren , Lampen & Sound	Ein/Aus, Bremsen oder Leistungsstufe 0..8	Drehrichtung vorwärts, rückwärts



Ereignis(se):	_____
Sensoren, Timer	

Zustandsname:	_____	
Aktoren Motoren , Lampen & Sound	Ein/Aus, Bremsen oder Leistungsstufe 0..8	Drehrichtung vorwärts, rückwärts

ARBEITSBLATT 5**FÜR ARBEITSGRUPPE:** _____**Bitte heute die Spezialistenfunktionen tauschen!**

Teammanager: _____ Informationsspezialist: _____

Materialspezialist : _____ Kommunikationsspezialist: _____

Aufgaben

Sucht euch für die nächsten Stunden eine Aufgabe aus, die ihr selbstständig im Team bearbeiten wollt. Tipps zur Konstruktion findet ihr in den Constructopédias. Im Internet findet ihr viele Adressen zum Thema Lego-MindStorms, z.B.

<http://www.legomindstorms.com/>

Eure Mitgliedsnummer lautet: AKGDKYE2ACJP
Die Seriennummer eines RCX 720995

<http://www.lego.com/dacta/home.asp><http://www.forwiss.uni-erlangen.de/MuT/bauanleitung.html>http://www.informatik.fh-hamburg.de/~lego/Projekte/body_projekte_deutsch.html<http://www.crynw.r.com/lego-robotics/><http://www.firstlegoleague.org/de>http://www.dragonslair.ch/project/WebSite_Robotics/doku/Roboter_HW.pdf

Ihr könnt auch eine Suchmaschine verwenden, um aktuelle Seiten zu finden.

Aufgabenstellungen

Änderungen am Roverbot:

- Caterpillar: Fahrzeug mit Kettenantrieb
- Käfer: Fahrzeug mit „6-Antriebsbeinen“
- Spezial: Wurfkatapult oder ähnliches mit 3. Motor

Neue Robotertypen:

- Mars-Lander: Plattform mit Rampe und Ladekran
- Mars-Rover: Geländegängiges Fahrzeug mit 3-Achsen
- Rennauto: Fahrzeug mit Differential und Zahnstangenlenkung
- Acrobot: Fahrzeug mit großen Rädern, das sich überschlagen kann.
- Inventorbot Stehender Roboter, der winken, grüßen und werfen kann.

MarsAufgaben:

- MarsCD installieren;
- Neue Challenges sind unter „LEKTIONEN“ → „AUFGABEN“ abrufbar

ANLAGE ZUM ARBEITSBLATT 5 FÜR LEHRERINNEN UND LEHRER

Programmcode: MARS01.LSC



Abbildungen:



Hauptprogramm(links oben),
Methode vorwaerts (mitte, oben),
Methode blinken10x (rechts, oben)
Methode runter (rechts unten)
Methode folgen (links, unten)



ANLAGE ZUM ARBEITSBLATT 5 FÜR LEHRERINNEN UND LEHRER

Programmcode: RENNWAGEN1.LSC



Abbildungen:

Hauptprogramm (links)
 Methode basis (links oben)
 Methode linksrum (mitte oben)
 Methode rechtsrum (rechts oben)
 Methode schnellvor (unten)