



IMST – Innovationen machen Schulen Top

Informatik kreativ unterrichten

COMPUTATIONAL THINKING – WAS LÄUFT IM INFORMATIKUNTERRICHT?

ID 1161

Birgit Höbbling

Mag. Dr. Ulrike Oberheber

Praxisschule Verbund Neue Mittelschule der PH Kärnten

BRG - BORG Waidmannsdorf

Klagenfurt, Juli 2014

INHALTSVERZEICHNIS

INHALTSVERZEICHNIS	2
ABSTRACT	3
1 EINLEITUNG	4
1.1 Motivation und Rahmenbedingungen des Projekts	4
1.2 Ziele	4
1.2.1 Ziele auf Schüler_innenebene.....	4
1.2.2 Ziele auf Lehrer_innenebene.....	5
1.3 Vorgangsweise.....	6
1.3.1 Gruppendesign.....	6
1.3.2 Phase 1 . Erarbeitung von Programmstrukturen.....	6
1.3.3 Phase 2 - Zusammenbau und Programmierung der Roboter.....	7
1.4 Zeitplan	7
2 PROJEKTIHALT	8
2.1 Projektstart	8
2.1.1 Das Projekt . Was Roboter können.....	8
2.1.2 Unser erstes Scratch-Programm.....	9
2.1.3 Bühne frei für Scratch und seine Freunde.....	12
2.1.4 Geometrische Figuren . Programmsequenz und Schleifen	13
2.1.5 Steuerung des Programmablaufs . Bedingte Anweisung.....	13
2.1.6 Zusammenbau der Roboter	15
2.1.7 Nummer 5 lebt.....	16
2.1.8 Einführung von Enchanting.....	18
3 EVALUATION	20
3.1 Evaluation projektspezifischer Ziele.....	20
3.1.1 Lernerfolg	20
3.1.2 Selbsteinschätzung des Lernerfolges.....	20
3.1.3 SWOT . Analyse	21
3.2 Evaluation aus Sicht der Ziele des Themenprogramms.....	22
3.3 Evaluation aus Sicht übergeordneter IMST Ziele	22
3.3.1 Gender- und Diversitätsaspekte.....	22
3.3.2 Schulentwicklungs- und Disseminationsaspekte	22
4 ZUSAMMENFASSUNG	23
5 LITERATUR	24
6 ANHANG	25

ABSTRACT

Computational Thinking führt uns in die Welt der Denk- und Konstruktionsprozesse eines digitalisierten Alltags. Dabei wird versucht einen Ausschnitt des Alltags zu modellieren. Dieser Prozess gliedert sich in drei Ebenen: Aktion, Abstraktion und Reflexion. Diese Charakteristika eines Problemlösungsprozesses begegnen uns im Informatikunterricht als Problemstellung – „*etwas läuft*“, als Suche nach passenden Algorithmen – „*wie können wir verstehen/darstellen was läuft*“ und als Prüfung der Tauglichkeit – „*ich weiß jetzt, wie es läuft und kann es vielleicht besser machen*“. Roboter bieten einen attraktiven Zugang zur Technik. Man kann durch Anfassen und Ausprobieren entsprechend adaptierter Roboter schon innerhalb weniger Stunden Grundkenntnisse der Konstruktion und Programmierung von Robotern erlernen. Dabei erfährt man, dass Technik Spaß macht und lernt, wie technische Systeme entwickelt werden und wie ein Computerprogramm abläuft. Technische Fertigkeiten werden beim Zusammenbau von Robotern trainiert und das Erarbeiten von Programmierkenntnissen erweitert den Handlungsspielraum. Motiviert durch Alltagsphantasien und Sciencefiction stellt sich jedoch auch die Frage, ob der technischen Machbarkeit selbst Grenzen unterworfen ist.

Schulstufe: 8. Schulstufe
Fächer: Informatik/ Wahlpflichtfach
Kontaktperson: Birgit Hölbling
Kontaktadresse: Hubertusstraße 1,
9020 Klagenfurt

1 EINLEITUNG

Die Projektidee wurde im Zuge der Vorbereitungen für den Tag der offenen Tür der Praxisschule Verbundmodell Neue Mittelschule Campus Hubertusstraße geboren. Die Schüler_innen des Wahlpflichtgegenstandes „Informatik“ begannen sich in das Thema der Programmierung einzuarbeiten. Vom handlungsorientierten Unterricht auf einfachster Stufe bis hin zur Abstraktionsebene beim Programmieren arbeiteten die Schüler_innen kooperativ und selbstständig.

1.1 Motivation und Rahmenbedingungen des Projekts

Dieses Projekt wird mit 24 Schüler_innen durchgeführt. Den Projektteilnehmer_innen stehen zwei Informatikräume mit insgesamt 24 Computern zur Verfügung. Zusätzlich gibt es sieben Baukästen mit Legorobotern (Lego NXT), die im Laufe des Projektes von den Schüler_innen zusammengebaut werden. Die Legoroboter werden mit Enchanting programmiert.

Durch den praktischen Umgang mit elektronischen und mechanischen Bauteilen soll das Interesse der Schüler_innen an den Fachbereichen Mathematik, Informatik, Naturwissenschaft und Technik (MINT – Fächer) geweckt werden. Der Einsatz von Robotern bietet eine für Schüler_innen attraktive Möglichkeit, zu sehen, wie ausgehend von der Frage, ob bestimmte Aufgaben nicht auch durch Maschinen erledigt werden können, ein umfangreicher Prozess in Gang gesetzt wird. Die Maschine muss gebaut werden, und dafür braucht es einen (guten) Plan. Die Maschine soll vorgegebene Abläufe ausführen und dafür braucht es eine (maschinenverständliche) Sprache – diese Sprache ist die Programmiersprache. Einen eigenen Roboter zu bauen und zu programmieren und gleichzeitig eine reflektierte Haltung zum Thema technische Machbarkeit zu erarbeiten, ist die Herausforderung des Projektes.

1.2 Ziele

Im Projekt „Computational Thinking – Was läuft im Informatikunterricht?“ ist das Ziel gewissermaßen Programm. Die Frage, kann man Computational Thinking im Informatikunterricht lernen, oder was lernt man dabei überhaupt - führt uns in die komplexen Zusammenhänge von Denk- und Konstruktionsprozessen in einer zunehmend digitalisierten Alltagswelt. Mit einem technischen Equipment im Hintergrund an Fragestellungen heranzugehen, führt zu neuen Strategien bei der Problemlösung. Zugleich ändert sich auch die Sichtweise auf Alltägliches.

1.2.1 Ziele auf Schüler_innenebene

1.2.1.1 Teilziel 1: Kennenlernen von Programmstrukturen

Die Schüler_innen sollen grundsätzliche Abläufe des Programmierens erarbeiten. Sie werden durch das Bearbeiten der gestellten Aufgaben grundlegendes Wissen zum Thema Programmierung kennenlernen. Die Erstellung von Sequenzen (Programmschritte), die Arbeit mit Wiederholungen (Schleifen), Entscheidungen (wenn, dann, sonst) und Interaktionen (Benutzereingaben) als Elemente eines Programmcodes sind inhaltliche Schwerpunkte der einzelnen Unterrichtseinheiten. Das Projekt ist so konzipiert, dass in jeder Arbeitseinheit die drei Ebenen Aktion, Abstraktion und Reflexion zusammenwirken. Durch das aktive Durchspielen von Befehlsabfolgen soll erreicht werden, dass die Umsetzung in den abstrakten Programmcode leichter gelingt. Die Planungsphase wird unabhängig vom Computer durchgeführt und so als wesentliches Element des Arbeitsprozesses definiert.

1.2.1.2 Teilziel 2: Interaktion zwischen Computer und Roboter verstehen

Die Schüler_innen sollen den Roboter mit Hilfe von Anleitungen selbstständig zusammenbauen und programmieren. Beim Programmieren des Roboters sollen sie auf das erarbeitete Programmierwissen aufbauen und es als Interaktion zwischen Computer und Roboter konkretisieren. Der Aktionsradius wird somit erweitert und die Schüler_innen können ihre Strategien in einem neuen „Umfeld“ testen. Zusätzliche Kompetenzen werden angesprochen und das Handlungsrepertoire wird erweitert. Die Arbeitsschritte: *Planung*, *Programmierung* und *Adaption* sowie *Präsentation*, werden um die Schritte *Roboter bauen - umbauen* und *Transfer testen - adaptieren* erweitert. Die zunehmende Komplexität erfordert, dass die Schüler_innen zusätzlich Dokumentationsschritte, die den Prozess sichtbar und kontrollierbar machen, verwenden.

1.2.1.3 Teilziel 3: Kritische Betrachtung von Robotern und der Interaktion Mensch - Roboter

Motiviert über Alltagsphantasien und Sciencefiction sollen die Schüler_innen einen altersadäquaten Blick auf die Chancen und Gefahren einer digitalisierten Welt (Computer – Roboter) werfen. Durch mediale Unterstützung (Film: „*Nummer 5 lebt*“) sollen sie auch auf die Leitfrage „*Können Computer alles?*“ in der Reflexionsphase kritisch eingehen. Sie sollen für Fragestellungen, die sich angesichts sensorgesteuerte Denkmaschinen ergeben, sensibilisiert werden.

1.2.2 Ziele auf Lehrer_innenebene

1.2.2.1 Teilziel 1: Erarbeitung von Planungskompetenzen

Das Ziel ist es, den Unterricht so zu gestalten, dass es den Schüler_innen möglich ist, selbstorganisiert an den Themen zu arbeiten. Die einzelnen Arbeitsphasen sollen einem klaren Ablaufkonzept folgen. Die Unterrichtsmaterialien werden so gestaltet, dass der Transfer theoretisch erarbeiteten Wissens und damit die geplanten Abläufe in der Programmierung gelingen. Hier soll die Zusammenarbeit der Gruppenteilnehmer_innen durch Formen des kooperativen Lernens unterstützt werden und ein konstruktives Team entstehen.

1.2.2.2 Teilziel 2: Persönliche Entwicklung fördern

Die Arbeitsaufgaben werden so gestaltet, dass unterschiedliche Kompetenzen angesprochen werden. Durch die Mitarbeit am IMST-Projekt soll der Selbstwert der Beteiligten gehoben und die Ausdauer gestärkt werden. Wir versuchen jede_n Schüler_in so in das Projekt einzubinden, dass jede_r auch etwas über sich selbst lernt. (Reflexionssequenzen am Ende der jeweiligen Arbeitseinheiten).

1.2.2.3 Teilziel 3: Knowhow Transfer

Hauptthema des Unterrichts ist es, Computational Thinking erlebbar zu machen. Hier werden die Projektphasen so gestaltet, dass sich die erarbeiteten Strategien auch außerhalb des Informatikunterrichts anwenden lassen. Der Einsatz des kooperativen Lernens soll auch in anderen Fächern bzw. Klassen stattfinden.

1.3 Vorgangsweise

1.3.1 Gruppendesign

Das Thema Computational Thinking wird im Projekt auf mehreren Ebenen bearbeitet. Drei Kontexte sind dabei bestimmend: *Aktion, Abstraktion, Reflexion*.

Die Arbeit der Schüler_innen ist in Gruppen organisiert. Die Gruppen finden sich in der Starteinheit und sollten sich im Laufe des Projektes zu *funktionierenden* Teams entwickeln. In jeder Projekteinheit (2 Schulstunden pro Woche) nehmen die Schüler_innen definierte Teamrollen wahr und geben ein Feedback zu ihrer individuellen und der Gruppenleistung ab. Dieses Gruppendesign unterstützt die Umsetzung der Methoden des kooperativen Lernens, die je nach Themenstellung eingesetzt werden.

Um die Schüler_innen zur aktiven Verarbeitung der angebotenen Inhalte zu motivieren, werden die Grundprinzipien des kooperativen Lernens¹ (Denken, Austauschen, Vorstellen) angewandt. Die Konstruktion wird jeweils durch kurze Inputs unterstützt. Die Instruktion durch Schüler_innen, die Gruppenergebnisse präsentieren, wird bei Bedarf durch weitere Inputs ergänzt. Die Selbstorganisation des Lernens wird in diesem Setting besonders gefördert. Die dahinterliegende konstruktivistische Lerntheorie sagt, „vielmehr muss jeder Schüler die ihm angebotenen Inhalte aktiv verarbeiten und in seine mentalen Strukturen, also in seine individuellen Wissensnetze integrieren.“² Die Querverbindungen zwischen Robotik und Artificial Intelligence können im Rahmen des Projektes nicht explizit behandelt werden. Bei der Diskussion über den Film „Nummer 5 lebt!“ wird die Verschränkung der Themen Lernen und „Computational Thinking“ von den Schüler_innen intuitiv erfasst. Nummer 5 „lernt aus seinen Fehlern, er kann nicht genug Input haben.“³

In jeder Gruppe gibt es folgende Rollen: Teamleader, Support, Dokumentation und Präsentation. Diese zugeteilten Rollen können in jeder Einheit innerhalb der Gruppe ausgetauscht werden, damit im Laufe des Projektes möglichst jede_r in allen Rollen tätig ist. Die Evaluation ergibt keine gravierenden Unterschiede in der Beurteilung der Projekteinheiten, die in Abhängigkeit von der Rolle gesehen werden kann.

In den Erarbeitungsphasen und während der Gruppenarbeiten sollen sich alle Projektteilnehmer_innen in einem Raum befinden. Während der Programmierphasen arbeitet jeweils die Hälfte der Schüler_innen in einem der zwei Informatikräume.

In der konkreten Durchführung zeigt sich, dass die Gruppengröße für die Arbeit mit den Robotern verringert werden muss. Die Vergabe von gleichwertigen Aufgaben für vier Schüler_innen mit (nur) einem Roboter, ist nicht möglich. Die Gruppengröße wird auf zwei Schüler_innen verringert, wobei die Zeit für die Arbeit mit den Robotern auf jeweils eine Einheit der Doppelstunde fixiert wird. In der zweiten Einheit wird an der Vertiefung der Programmierkenntnisse gearbeitet. Die Aufgabenstellungen sind weitgehend unabhängig voneinander. Ein Transfer neu erarbeiteten Programmierwissens in die Programmierumgebung Enchanting wird jedoch angestrebt.

1.3.2 Phase 1 – Erarbeitung von Programmstrukturen

Kernthema der ersten Phase ist die Erarbeitung von Programmstrukturen.

- **AKTION** (manuelle/praktische Ebene)
Die Schüler_innen erarbeiten mittels Bausteinen grundlegende Programmstrukturen und Algorithmen (Sequenz, Wiederholung, ...).

¹ Ludger Brüning/Tobias Saum; Erfolgreich unterrichten durch Kooperatives Lernen; Neue Deutsche Schule Verlagsgesellschaft mbH; Essen, 2009. S. 17.

² Ebd. S. 21

³ Schülerkommentar; siehe weiter unten im Text: 2.1.7. Nummer 5 lebt

- **ABSTRAKTION** (Ebene der programmtechnischen Umsetzung)
Die erarbeiteten Strukturen werden am Computer umgesetzt (einfache Scratch-Programme werden geschrieben und präsentiert).
- **REFLEXION** (kritische/reflexive Ebene)
Gibt es im Alltag Abläufe, die wie ein Programm funktionieren? Mein inneres Programm – Warum ich kein Roboter bin!

1.3.3 Phase 2 - Zusammenbau und Programmierung der Roboter

Kernthema der zweiten Phase ist der Zusammenbau und die Programmierung eines Roboters.

- **AKTION** (manuelle/praktische Ebene)
Ein Lego-Mindstorm Bausatz wird zusammengebaut. Wissen über Bauteile erarbeitet.
- **ABSTRAKTION** (Ebene der programmtechnischen Umsetzung)
Ein Roboter spaziert durch unsere Klasse. Er führt ein Programm aus und „reagiert“ auf seine Umgebung.
- **REFLEXION** (kritische/reflexive Ebene)
Welche programmierbaren Geräte umgeben uns? Was können Roboter? Ich weiß wie´s läuft!
Der Mythos der Machbarkeit!

1.4 Zeitplan

September 2013

Teilnahme an der IMST-Tagung mit Planungsworkshop
Detailplanung und Erstellung von Unterrichtsmaterialien

Oktober 2013

Start des Projektes
Einstieg in die Thematik; Erste Schritte in der Programmierumgebung von Scratch

November 2013

Erarbeitung der grundlegenden Programmstrukturen
Festigung der Arbeitstechnik
Planung → Codierung → Testphase → Adaptierung → Präsentation
Umsetzung eines Programmierprojektes

Dezember 2013

Zusammenbau der Roboter
Roboter Projekte umsetzen

Jänner 2014

Präsentation der Projekte am Tag der offenen Tür der Praxisschule
Verbundmodell Neue Mittelschule Campus Hubertusstraße
Projektfinale

Februar 2014

Evaluation des Projekts

2 PROJEKTINHALT

Das Projekt „Computational Thinking – Was läuft im Informatikunterricht“ ist mittlerweile erfolgreich abgeschlossen. Der Start des Projektes erfolgte Mitte November und es wurde zum Ende des Wintersemesters beendet. Schon während der Projektdurchführung kamen formative Evaluationsmethoden zum Einsatz. Diese wurden am Ende jeder Projekteinheit schriftlich und anonym von den Schüler_innen abgegeben. So wurde jede Projekteinheit mit Feedbackbögen (siehe Anhang 14) evaluiert. Sowohl allgemeine wie auch inhaltliche Aspekte wurden dabei abgefragt. Am Ende des Projektes erfolgte eine SWOT-Analyse aus Schüler_innenperspektive. Außerdem wurde eine Befragung mittels Fragebogen durchgeführt.

2.1 Projektstart

Der Projektstart erfolgt im November. Die Schüler_innen werden informiert, dass sie Teilnehmer_innen eines IMST-Projektes sein werden. Sie erhalten Informationen zum Projektablauf und zur Arbeitsweise. Die anfänglich 24 Schüler_innen werden in sechs Gruppen eingeteilt und bekommen eindeutige Rollen in den jeweiligen Gruppen zugewiesen.

2.1.1 Das Projekt – Was Roboter können

Die erste Projekteinheit enthält folgende Arbeitsaufträge:

- “ Überlege alleine was Computer - Roboter alles können.
- “ Notiere deine wichtigste Überlegung in einem Satz auf deinem Arbeitsblatt!
- “ Suche in der Klasse möglichst viele Mitschüler_innen, die deine Ansicht mit ihrer Unterschrift unterstützen. Wer unterschreibt, muss die Überlegung wiederholen können!

Um diesen Arbeitsauftrag ausführen zu können, erhalten die Schüler_innen ein Thesenblatt (Anhang 1), das sie gemeinsam bearbeiten. Die Ergebnisse werden in einer Power Point Präsentation direkt im Anschluss gesammelt und präsentiert.

Danach erfolgt die Gruppeneinteilung, wobei die Schüler_innen sich selbst aussuchen können, in welcher Gruppe sie arbeiten möchten. Insgesamt arbeiten 6 Gruppen zu je 4 Personen am Projekt mit. Danach erfolgt eine genaue Anweisung für das Arbeiten in den Gruppen in Form von einem Flussdiagramm. Diese Form der Darstellung wird bewusst gewählt. Das Flussdiagramm als Darstellungsmethode für Programmläufe lässt sich gut für die Darstellung von Arbeitsabläufen einsetzen. So kann den Schüler_innen mit einem kurzen Input gezeigt werden, wie Methoden aus der Programmierung in anderen Zusammenhängen hilfreich eingesetzt werden können.

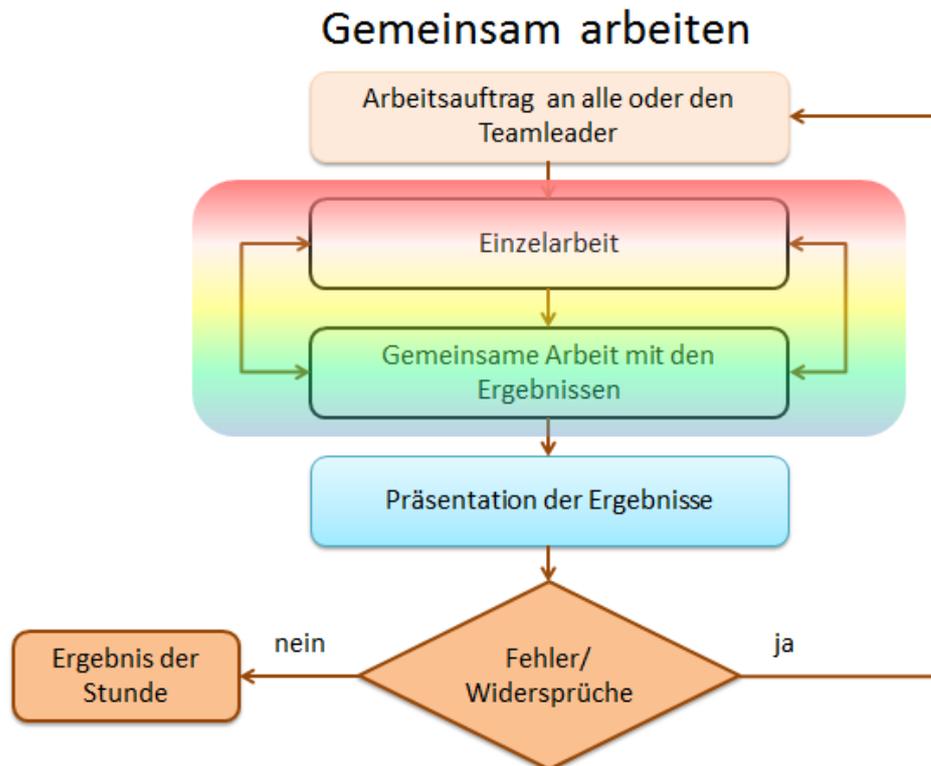


Abbildung 1: Flussdiagramm – Kooperatives Lernen

2.1.1.1 Evaluation der ersten Projekteinheit

	☺	☹	☹☹
Die heutige Einheit war:	17	4	1

In dieser ersten Projekteinheit gingen die Schüler_innen mit Neugierde und Freude an das Thema heran. Einige waren noch etwas skeptisch, was die „neue Unterrichtsform“ anbelangte. Ein Schüler war mit seiner Gruppe nicht zufrieden. Dieser Schüler verließ jedoch im Laufe des Schuljahres die Klasse und somit wurde das Projekt mit 23 Schüler_innen weitergeführt.

2.1.2 Unser erstes Scratch-Programm

In dieser Einheit unseres Projektes lautet die wesentliche Fragestellung „Was ist ein Computerprogramm?“. Die Schüler_innen sollen sich eingehend mit folgenden Aussagen auseinandersetzen.

- “ Überlegt einzeln, wie ihr erklären würdet, was ein Computerprogramm ist!
- “ Einigt euch in der Gruppe auf eine Erklärung!
- “ Der Präsentator trägt die Gruppenmeinung vor.

In den einzelnen Gruppen werden die Themen diskutiert und in Form einer Projektmappe dokumentiert. Die Ergebnisse und Erklärungen müssen dann vom Gruppenpräsentator den anderen Projektteilnehmer_innen vorgestellt werden.

Kernaussage: Ein **Computerprogramm** oder kurz **Programm** ist eine den Regeln einer bestimmten Programmiersprache genügende **Folge von Anweisungen**, um auf einem Computer eine bestimmte Aufgabe bearbeiten zu können.

Danach folgt die Vorstellung der Katze von Scratch (**Scratch 1.4 für Windows**), als programmierbares Objekt. Anfänglich übernehmen einzelne Schüler_innen die Rolle der Scratch - Katze und führen die Anweisungen der Gruppenmitglieder aus. Der Arbeitsauftrag lautet:

- **Bühne frei:** Als Akteur_in führst du Anweisungen genauso aus, wie sie dir gegeben werden. Du kannst leider keine eigenen Ideen einbringen. Die Programmierer_innen sind verantwortlich, dass das Programm funktioniert.

Damit sich die Schüler_innen handlungsorientiert dem Thema Programmierung annähern können, erhalten Sie Scratch-Programmierblöcke aus Papier und legen die Anweisungen in Form eines Programmpuzzles auf.

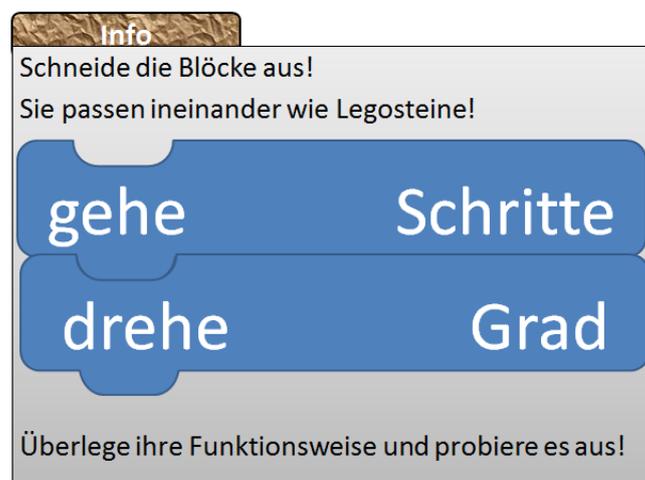


Abbildung 2: Programmierbausteine Scratch

Danach folgen konkrete Arbeitsanweisungen:

- “ Erstellt in der Gruppe eine Abfolge, die einen Weg rund um euren Gruppentisch beschreibt.
- “ Vergesst nicht, dass ein Startpunkt gebraucht wird.
- “ Lasst die Nachbargruppe euer Programm testen.

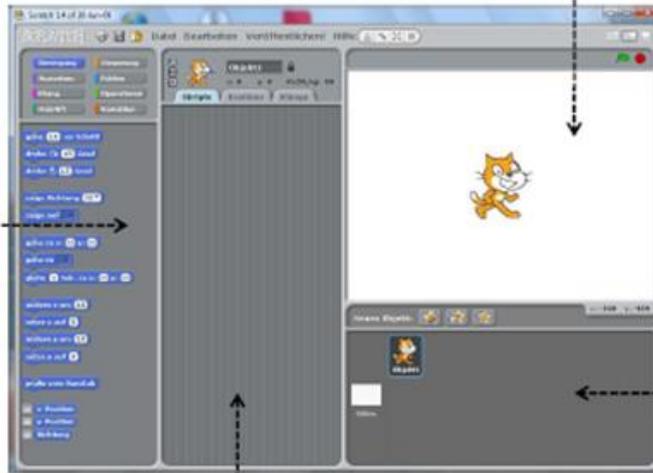
Damit die Schüler_innen verstehen, warum sie diese Programmierblöcke benötigen, wird ihnen Scratch vorgestellt. Scratch (**Scratch 1.4 für Windows**) bietet eine Programmieroberfläche, die es ermöglicht eigene interaktive Geschichten, Animationen, Spiele und Musikwerke zu erstellen und sie als Scratch-Projekte anderen über das Internet zur Verfügung zu stellen. Die Benutzeroberfläche wird in vier Bereiche eingeteilt:

- “ Die Blockpalette
- “ Die Bühne
- “ Die Objektleiste
- “ Der Programmierbereich

Die Benutzeroberfläche

Bühne: Hier läuft dein *Projekt* (Geschichte, Animation, Spiel, Musik...) ab. Die handelnden Lebewesen und Gegenstände in deinem Projekt heißen *Objekte*.

Blockpalette: Von hier holst du die Blöcke, aus denen du dann im Programmierbereich dein Programm zusammenbaust.



Objektliste: Um ein Objekt zu programmieren, wählst du es hier aus. Am Anfang gibt es nur ein Objekt, die Katze Scratch.

Programmierbereich: Hier baust du Blockstapel, die den Objekten sagen, was sie tun sollen. Die Stapel aller Objekte zusammen bilden dein *Programm*.

Abbildung 3: Benutzeroberfläche Scratch

Quelle: <http://www.kinderlabor.ch/cms/upload/Scratch-Unterlagen/Scratch-Handbuch.pdf>

Danach folgen folgenden Arbeitsaufträge:

- “ Öffne das Programm Scratch auf deinem Computer und schau dir die Oberfläche genau an.
- “ Lass Scratch ein Rechteck zeichnen!
- “ Verwende die bekannten Blöcke und zeige dein Programm deinen Gruppenmitgliedern!
- “ Ein Programm pro Gruppe wird präsentiert!
- “ Speichere dein Programm im Projektordner!

Abschließend erfolgt eine angeleitete Reflexion zum Thema – *Programme in unseren Köpfen*:

- “ Gibt es Situationen, in denen du selbst wie ein Programm funktionierst?
- “ Gibt es Anweisungen, die du befolgst wie ein Roboter?
- “ Diskutiert in der Gruppe und findet ein Gruppenstatement!
- “ Die/der Präsentator_in stellt die Gruppenmeinung vor!

2.1.2.1 Evaluation der zweiten Projekteinheit

	😊	😐	☹️
Die heutige Einheit war:	21	1	
Ich kenne Bewegungsbausteine und kann sie zusammensetzen:	20	2	

In dieser Projekteinheit waren 2 Schüler_innen nicht anwesend. Nach der anfänglichen Unsicherheit einiger Projektteilnehmer_innen in der ersten Projekteinheit, waren in dieser Unterrichtsstunde alle mit Interesse und Freude dabei. Das „gegenseitige Programmieren“ wurde diszipliniert und motiviert durchgeführt. Auch die Zusammenarbeit in den jeweiligen Gruppen erfolgte reibungslos.

2.1.3 Bühne frei für Scratch und seine Freunde

Das Bühnenbild dient als Hintergrund für alle Objekte, die sich in einem Programm bewegen. So gelingt es Geschichten zu erzählen. Bewegte Geschichten nennt man Animationen. Die Katze - Scratch ist ein Objekt, das beliebig gestaltet werden kann.

Wird ein Objekt in verschiedenen Ausführungen gebraucht, so kann man Kostüme entwerfen, um das Objekt z. Bsp. an eine Szene anzupassen. In dieser Projekteinheit sollen die Schüler_innen folgende Arbeitsaufträge ausführen.

- “ Öffne ein neues Projekt und klicke auf das Symbol Bühne.
- “ Nun können unter dem Karteireiter Hintergründe neue Hintergründe gezeichnet oder importiert werden.
- “ Gestalte mindestens einen neuen Hintergrund. Der erste, leere Hintergrund kann gelöscht werden.
- “ Lösche Objekt1 (Katze) und füge eigene Objekte durch Auswahl aus den bestehenden Objekten hinzu oder kreierte ein neues Objekt, das auf deine Bühne passt.
- “ Am Ende sollte eine Ferienlandschaft mit ausgewählten Objekten entstehen.

2.1.3.1 Evaluation der dritten Projekteinheit

	😊	😐	☹️
Die heutige Einheit war:	22	2	
Ich kann Scratch aufrufen und ein Objekt auf einer Bühne bewegen:	23	1	

In dieser Einheit konnten die Schüler_innen zum ersten Mal „wirklich programmieren“. Das heißt, sie nutzen die Möglichkeiten der Programmieroberfläche von Scratch. Da es sich hierbei um eine grafisch ansprechende Programmiersprache handelt, waren auch alle begeistert davon, wie einfach und schnell sich ein Objekt bewegen lässt.

2.1.4 Geometrische Figuren – Programmsequenz und Schleifen

Die Schüler_innen sollen in dieser Einheit Programme erstellen, deren Struktur über die einfache Abfolge von Programmschritten hinausgeht. Die Sequenz wurde bereits erarbeitet, nun soll die Wiederholung von zusammengehörenden Programmschritten als Schleife programmiert werden. Hier beginnen wir mit dem Zusammenfügen von Scratchbausteinen, die man zum „Zeichnen“ eines Dreiecks benötigt. Damit man nun von der einfachen geometrischen Konstruktion eines Dreiecks zu einem schönen Dreiecksmuster, das aus gedrehten Dreiecken gebildet wird, gelangt, wird die Wiederholung als Programmschleife eingeführt. Die Schüler_innen erkennen, dass durch die Angabe der nötigen Programmschritte und der Wiederholungszahl mit wenigen Anweisungen ein komplexer Ablauf bzw. ein komplexes Bild entstehen kann. Die Bausteine der Kategorie *Steuerung* werden eingeführt und ihre Funktionsweise getestet.

Folgende Arbeitsaufträge sollen in der Gruppe ausgeführt werden:

- “ Erstelle mit deinen Bausteinen eine Sequenz, mit der ein Dreieck gezeichnet wird.
- “ Überlege, wie eine Sequenz zum Zeichnen eines Kreises mit deinen Bausteinen erzeugt werden kann! Gibt es Programmteile, die immer gleich bleiben?
- “ Nun wollen wir eine Blume zeichnen, die aus Viertelkreisen besteht! Beginne mit einem Viertelkreis!

2.1.4.1 Evaluation der vierten Projekteinheit

	☺	☹	☹
Die heutige Einheit war:	24	0	
Ich kann Bühnen und Objekte gestalten:	20	4	
Ich kenne die Begriffe Algorithmus, Sequenz und Schleifen:	8	16	

Da wir uns schon mitten im Projekt befanden, war das Arbeiten in den einzelnen Gruppen mittlerweile automatisiert. Das heißt, jeder wusste was zu tun war, kannte seine Aufgaben und die Gruppen organisierten sich größtenteils selbst. Das wirkte sich auch auf das Arbeitsklima gut aus. Da aber auch die Arbeitsaufträge komplexer und schwieriger wurden, waren einigen Projektteilnehmer_innen leicht überfordert.

2.1.5 Steuerung des Programmablaufs – Bedingte Anweisung

Um den Ablauf eines Programms zu kontrollieren, verwendet man die **bedingte** Anweisung. Dabei wird zuerst eine Bedingung formuliert z. Bsp. wenn eine bestimmte Taste gedrückt wird. Auf diese Bedingung muss das Programm jederzeit reagieren können. Scratch verwendet dafür einen sogenannten **HUT**, der die Bedingung enthält!

Soll innerhalb einer Sequenz eine Entscheidung getroffen werden, so verwendet man den Steuerungsblock „falls“. Abhängig davon, ob die Bedingung erfüllt ist, erfolgt die gewünschte Aktion. Auch für den Fall, dass die Bedingung nicht erfüllt ist, kann eine Aktion festgelegt werden.

Den Schüler_innen wird als Anleitung für den Entwurf von bedingten Anweisungen ein Flussdiagramm gezeigt.

Der Arbeitsauftrag in dieser Projekteinheit lautet:

- “ Öffne Scratch und erzeuge ein beliebiges Objekt.
- “ Das Objekt soll in der Mitte der Bühne platziert werden und nach rechts schauen.
- “ Nun soll das Objekt durch Tasteneingaben des Benutzers gesteuert werden.
- “ Bedingung formulieren:
Wird die Taste ↑ gedrückt,
- “ Ereignis festlegen:
..., dann soll sich das Objekt nach oben drehen und sich in diese Richtung bewegen!

Anschließend soll nun folgender Ablauf programmiert werden:

Falls Scratch auf den oberen Rand trifft, soll sie „verschwinden“ und am unteren Rand wieder auftauchen. Dafür verwenden wir eine sogenannte vollständige Alternative. Es wird eine Bedingung mit zwei alternativen Reaktionen programmiert. Das kannst du dir so vorstellen.

- “ falls ... eine Bedingung eintritt
- “ dann ... erfolgt eine bestimmte Reaktion
- “ sonst erfolgt eine alternative Reaktion

Als Erweiterung des Befehlssatzes wird der Befehle *senden an alle* erarbeitet.

Folgender Arbeitsauftrag:

Entwirf eine Bühne mit Hindernissen. Platziere dein Objekt auf der Bühne. Es soll sich auf der Bühne entsprechend der Tasteneingabe des Benutzers bewegen. Falls ein Hindernis berührt wird, erscheint eine Warnung und das Spiel beginnt von vorne!

2.1.5.1 Evaluation der fünften Projekteinheit

	😊	😐	☹️
Die heutige Einheit war:	12	5	
Ich kann ein Objekt mit Hilfe der Pfeiltasten steuern:	13	4	
Ich kann eine Bedingung formulieren:	11	6	

Insgesamt waren die Schüler_innen noch immer motiviert am Thema Programmierung beteiligt. In den einzelnen Gruppen gab es unterschiedliche Herangehensweisen an die Arbeitsaufträge. Manche Schüler_innen lehnten sich zurück und übernahmen die Ergebnisse der anderen. .

2.1.6 Zusammenbau der Roboter

In dieser Projekteinheit sollen die Schüler_innen in den jeweiligen Gruppen ihre Legoroboter zusammenbauen. Diese Roboter wurden aus der Produktserie des dänischen Spielwarenherstellers Lego bezogen. Das Kernstück der Produktserie ist ein Steuerungscomputer mit dem Namen **Mindstorms (NXT)**. Er besitzt Anschlüsse für mehrere Sensoren und Aktoren, sowie USB- und Bluetooth-Schnittstellen. Folgend sieht man die Funktionen des NXT-Bausteins im Überblick:

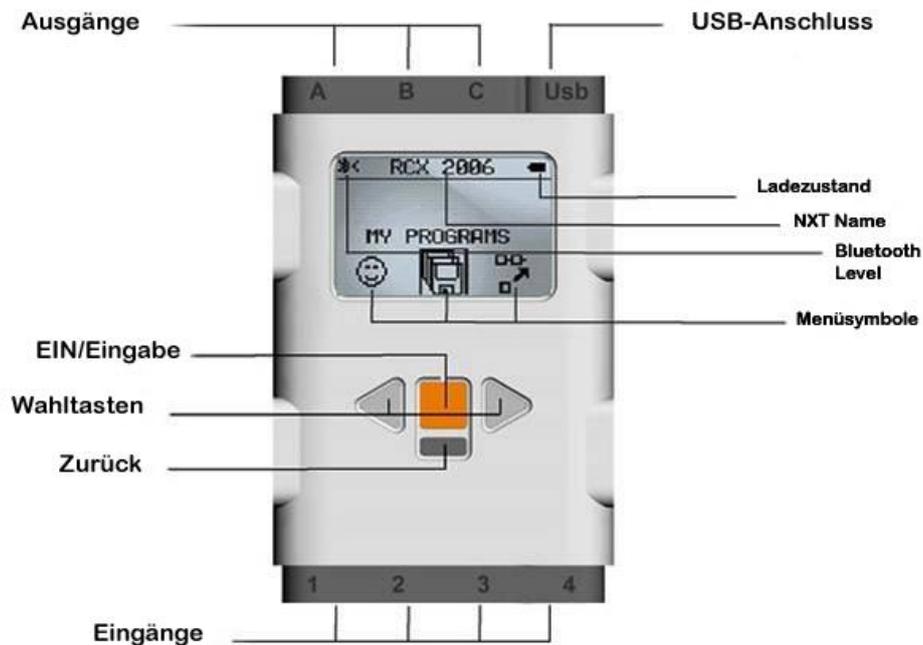


Abbildung 4: NXT – Baustein

Quelle: Lücking 2009: S.10

Außerdem gibt es Elektromotoren, Sensoren und Lego-Technik-Teile (Zahnräder, Achsen, Lochbalken, Pneumatik-Teile usw.), die an den Legoroboter angebaut werden können.⁴ Der Computer kann über spezielle Programmiersprachen am PC programmiert werden.

Für das Programmieren mit Schüler_innen der Neuen Mittelschule bietet sich die von Lego zur Verfügung gestellte NXT Programmieroberfläche an. Hierbei handelt es sich um eine graphische Oberfläche, die ein einfaches und übersichtliches Programmieren ermöglicht. Für die Durchführung des Projektes wird aber die auf Scratch basierende Programmiersprache Enchanting verwendet. Da es sich hierbei um Open Source Software handelt, ist auch keine Lizenz notwendig.

Zusammenbau der Roboter:

Jede der 6 Gruppen erhält den Legobaukasten (Lego NXT). Da jedes Gruppenmitglied eindeutige Aufgaben in der Gruppe hat, gibt es keine Diskussionen zur Aufteilung der Arbeit. Die Schüler_innen arbeiten hier selbstorganisiert und genau nach dem Bauplan der Legobaukästen. Jede Gruppe erhält den gleichen Bauplan und alle bauen vorerst denselben Roboter. Später können die Roboter nach den Vorstellungen der Schüler_innen umgebaut werden.

⁴ vgl. http://de.wikipedia.org/wiki/Lego_Mindstorms

2.1.6.1 Evaluation der sechsten Projekteinheit

	😊	😐	☹️
Die heutige Einheit war:	17		
Der Zusammenbau des Roboters hat mir gut gefallen:	17		
Ich weiß, worauf ich beim Zusammenbauen achten muss:	16	1	

In dieser Einheit war es sehr interessant zu beobachten, wie unterschiedlich die 6 Gruppen an das Zusammenbauen herangingen. Es kristallisierte sich bald heraus, dass zwei Gruppen sehr gut arbeiteten, zwei Gruppen durchschnittliche und zwei Gruppen schwächere Leistungen erbrachten. Auch die Herangehensweise der Mädchen war sehr interessant. Sie ließen sich gerne beim Zusammenbau von den Burschen helfen, ließen sich aber dennoch nicht die Arbeit aus der Hand nehmen.

2.1.7 Nummer 5 lebt

Um die Schüler_innen auch zu einer kritischen Auseinandersetzung mit dem Thema „Können Computer alles?“ zu animieren, sehen wir uns gemeinsam den Film „Nummer 5 lebt!“ an.

Hierbei handelt es sich um einen US-amerikanischen Science-Fiction-Film aus dem Jahr 1986. Auf einem amerikanischen Militärgelände werden neue Kampfroboter getestet. Als plötzlich ein Gewitter aufzieht, wird einer der Roboter vom Blitz getroffen. Nr. 5 ist von nun an kein Militärroboter mehr, sondern gerät völlig außer Kontrolle und verschwindet vom Gelände. Er findet Unterschlupf bei Stephanie, einer jungen Tierliebhaberin, und macht große Veränderungen durch: Auf einmal empfindet er Neugierde und Angst, kann über Witze lachen und hat gar keine Lust, sich auf Befehl selbst auszuschalten. Schließlich verliebt sich Nr. 5 auch noch in die hübsche Stephanie und löst dadurch einen ganz schönen Wirbel aus. Diese hat aber inzwischen das Militär und seine Wissenschaftler über seinen Verbleib informiert. Nr. 5 schwebt in größter Gefahr, denn seine "Erzeuger" haben nur die Zerstörung des außer Kontrolle geratenen Roboters im Sinn⁵.

Einen Science Fiction (SF) Film auszuwählen, hatte den Hintergrund, dass „SF-Filme – zumindest auf visueller, oftmals aber auch auf narrativer Ebene – das Epizentrum filmischer Artikulation von Technik und Innovation“⁶ darstellen. Filme über Roboter eignen sich in besonderer Weise zur Darstellung der unterschiedlichen Haltungen gegenüber technischer Innovationen und bieten einen guten Ausgangspunkt für Diskussionen mit den Schüler_innen. So ließ sich erkennen, dass die von Ruge dargestellte Genealogie der Roboterdarstellung im SF-Film von den Schüler_innen in ihren Beispielen und Vermutungen über das technisch Machbare eingebracht wurde. Hier sei die Kategorie Menschenähnlichkeit angeführt, die die Schüler_innen mit großer emotionaler Beteiligung besprochen wurden. Einige der von Ruge vorgeschlagenen Subkategorien wurden in der Diskussion über den Film angesprochen: „Bewusstsein, Lernfähigkeit, Willensstatus, visuelle Ähnlichkeit, Emotionalität“⁷. Sowohl „die-

⁵ vgl. <http://www.amazon.de/Nummer-5-lebt-Ally-Sheedy/dp/B00008CGZ7>

⁶ Ruge Wolfgang (2012). Roboter im Film Audiovisuelle Artikulationen des Verhältnisses zwischen Mensch und Technik. Magdeburger Schriftenreihe zur Medienbildung, Band 1. ibidem Verlag, Stuttgart. S. 36

⁷ Ebd. S. 78.

nende Einzelstücke“, wie auch „bedrohliche Massen“ von Cyborgs wurden genannt, wobei eine ausführliche Besprechung der einzelnen Kategorien in diesen Unterrichteinheiten nicht angestrebt wurde. Als Evaluation wurde für diese Einheit ein inhaltliches Arbeitsblatt gewählt (siehe Anhang 12). Die Texte der Schüler_innen ließen erkennen, dass eine Vielzahl von Aspekten in der Diskussion aufgegriffen wurde. Anstelle einer Zusammenfassung möchten wir unsere Schüler_innen zu Wort kommen lassen.

2.1.7.1 Evaluation der siebenten Projekteinheit

Mit dem Arbeitsblatt wurden Einschätzungen und die Meinungen der Schüler_innen erfragt. Die Texte der Schüler_innen wurden geringfügig geändert, um den Zusammenhang mit den Themen erkennbar zu halten.

- Einen Roboter (der Kategorie AIBO) „finde ich voll cool“, weil „er selber denkt“.
- Ein Roboter ist wie ein Computer, weil er ein technisches Gerät ist und einen kleinen Computer eingebaut hat.
- „AIBO ist gut für Leute, die eine Hundeallergie haben.“
- Der Roboter wird mit einem Toaster verglichen: „... ein Roboter, der zum Zeitvertreib gedacht ist – also ein Spielzeug.“
- „Als Toaster wird er (der Roboter) eher nix ...“
- Roboter sind wie Waschmaschinen: „Denn man könnte einen Roboter im Alltag gut gebrauchen, dass er Sachen für einen erledigt.“ Eine weiteres Statement zu diesem Vergleich: „Ich finde manchmal ist die Waschmaschine genauso kompliziert zu bedienen, wie den Roboter“.
- „Es war eine dumme Idee“ die Produktion von AIBO wieder einzustellen.
- AIBO dient einerseits der Forschung andererseits als Haustier.
- „Roboter sind wie Computer, weil der Computer genauso wie der NXT-Roboter programmiert werden kann, und weil der Roboter vom Computer programmiert wird.“
- „Ein Roboter ist mehr wie eine Konsole, weil er Touchsensoren und Spracherkennung hat.“
- „Roboter (auch AIBO) sind interessant, weil man „ihm Aufgaben im Haushalt erledigen lassen kann.“
- „AIBO ist interessant aber teuer und sinnlos, solange es echte Hunde gibt.“

Die Anmerkungen zur Verwandlung von Nummer 5 wurden von den meisten Schüler_innen als Kommentar des Gesehenen gestaltet. Hier einige Auszüge aus den Texten der Schüler_innen:

- Besonders an Nummer 5 ist, dass er „Reden kann – Computer können nicht reden. Er hat Gefühle und kann reden, tanzen, springen und Spiele spielen“.
- „Er kann eigenständig denken und ist schnell lernfähig.“
- „Er hat Gefühle. Roboter können sonst nur mit einem Programm arbeiten.“
- „Er kann sich selbst verändern.“
- „... und er hat sich selbst programmiert. Ich finde das besonders schön.“
- „Er ist durchgedreht und hat einen eigenen Willen entwickelt.“
- „Er hat Gefühle, kann reden und sich selber programmieren. Das ist besonders, denn kein Roboter kann sich von selber programmieren.“
- „Er hat angefangen zu leben und zu fühlen wie ein Mensch. Er lebt!“
- „... und lernt aus seinen Fehlern, er kann nicht genug Input haben.“
- „Er kann sich frei ohne eine Programmierung bewegen. ...“

2.1.8 Einführung von Enchanting

Da das Programm Enchanting (Enchanting v0.2.4.1) von den Programmierern von Scratch zur Steuerung von Robotern entwickelt wurde, ist die Einführung dieses Programmierwerkzeuges ohne großen Aufwand möglich. Wesentlich ist hier die Definition der Motoren und Sensoren. Zuerst müssen die jeweiligen Sensoren festgelegt werden, dazu erhalten die Schüler_innen die Enchanting Cards (siehe Anhang 10). Hier finden sie genaue Anleitungen, wie die Sensoren festgelegt werden und wie man einfache Programmstrukturen auf den Roboter übertragen kann. Folgende Fragestellung soll bearbeitet werden:

Wie können wir Enchanting mitteilen, dass am NXT der Lichtsensor an Port 3 angeschlossen ist?

- “ Klicke auf das “Fühlen” – Menü.
- “ Klicke auf “Sensoren festlegen”.
- “ Suche auf der linken Seite den Block mit den Lichtsensoreinstellungen.
- “ Ziehe ihn mit Hilfe zu „Sensor Port 3“.
- “ Wähle, welchen Sensor du verwendest: Lichtsensor, Farbsensor,
- “ Jetzt stehen dir die Lichtsensorenblöcke zur Verfügung.

Die weiteren Unterrichtseinheiten werden analog zu den Enchanting Cards aufgebaut. Die einzelnen Projektgruppen können individuell arbeiten.

Durch den Einsatz von Legorobotern im Informatikunterricht ändert sich auch das typische Unterrichten. Die Schüler_innen können sich in Gruppenarbeiten gegenseitig unterstützen und werden angeregt sich selbsttätig und eigenverantwortlich Wissen anzueignen und anzuwenden. Durch dieses selbstständige Lernen lernen die Schüler_innen „informell“. Das heißt, wer einem Problem nachgeht und versucht, es zu lösen, lernt informell. Dabei lernen sie teils bewusst, teils unbewusst. Es wird in den jeweils bestehenden sozialen, kommunikativen oder auch Arbeitszusammenhängen gelernt. Informelles Lernen bezeichnet ein Lernen in Lebenszusammenhängen, das ursprünglich vor allem als ein Lernen außerhalb des formalen Bildungswesens angesehen wurde. In den letzten Jahren hat die Diskussion zum informellen Lernen stetig an Bedeutung gewonnen und wird aktuell in fast allen pädagogischen Bereichen thematisiert.⁸

Informelles Lernen trägt wesentlich zu den Lernerfahrungen von Schüler_innen bei. Die Handy- und Computernutzung sind Beispiele für informelles Lernen. Jugendliche sind in beiden Gebieten vielen Erwachsenen überlegen. Wichtig für solche Lernerfolge ist die Lernmotivation, das heißt, der Wille, ein Gerät zu beherrschen und nutzen zu können. Die sofortige Rückmeldung durch das Gerät unterstützt den Lernprozess.

Auch hier knüpft das Programmieren von Legorobotern an. Durch das Zusammenbauen und Bedienen der Roboter und die direkte Rückmeldung von Programmbefehlen, werden die Schüler_innen motiviert und erweitern ihre Kenntnisse der Programmiersprache durch ausprobieren (trial and error). Hier wird versucht das formale Lernen, welches in Bezug auf Lernziele, Lernzeit oder Lernförderung stark strukturiert ist und zur Zertifizierung führt, durch ein informelles und nicht formales Lernen zu ersetzen. Formales Lernen ist aus der Sicht des Lernenden zielgerichtet. Beim Programmieren der Lego Roboter streben die Schüler_innen sicher auch das Ziel an, dass ihre Programme funktionieren, es ist jedoch nicht der strukturierte Ablauf und die Zertifizierung wichtig, vielmehr geht es da-

⁸ vgl. http://de.wikipedia.org/wiki/Informelles_Lernen

rum, dass die Schüler_innen mit Freude und Begeisterung Neues entdecken und dadurch komplizierte Vorgänge und Programmerroutinen verstehen und erforschen.⁹

2.1.8.1 Evaluation der achten Projekteinheit

	😊	😐	☹️
Die heutige Einheit war:	18	4	1
Ich kenne die Bausteine von Enchanting, um Sensoren anzusteuern:	14	9	
Ich kann ein Programm auf den Roboter übertragen und starten:	18	5	

Da diese Projekteinheit stark durch kooperatives und informelles Lernen geprägt war, haben sich einige Schüler_innen in diesen Lernstrukturen nicht so gut zurechtgefunden. Durch das eigenständige Arbeiten in der Zweiergruppe waren manche mit den Aufgabenstellungen überfordert und waren so schnell frustriert. Leider hat manchmal das Übertragen der Programme vom Computer auf die Roboter nicht funktioniert und damit stieg die Frustration. Nichtsdestotrotz hatten am Ende der Stunde alle einen „fahrenden Roboter“ und somit ein Erfolgserlebnis.

⁹ vgl. http://de.wikipedia.org/wiki/Informelles_Lernen

3 EVALUATION

3.1 Evaluation projektspezifischer Ziele

3.1.1 Lernerfolg

Am Ende des Projektes wurde ein Wissenstest mittels eines Lückentextes (siehe Anhang) und einer zusammenfassenden Programmieraufgabe durchgeführt. Die Programmieraufgabe war kompetenzorientiert gestaltet und umfasste mehrere Teilschritte, die jeweils steigende Kompetenzen erforderten.

Die Testergebnisse waren durchwegs positiv – in Noten: von Sehr gut bis Genügend. 3 Schüler_innen fehlten wegen Krankheit.

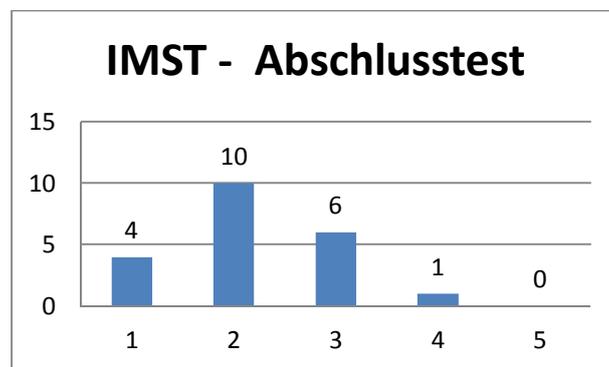


Abbildung 5: IMST - Abschlusstest

3.1.2 Selbsteinschätzung des Lernerfolges

Die Schüler_innen wurden nach ihrer Einschätzung über den Erfolg ihres Lernens während des Projektes gefragt. Sie haben auf folgende Fragen mit „ja“, „etwas“ oder „nein“ geantwortet.

- Das Nachdenken über Computerprogrammierung und Roboter hat mich neugierig auf dieses Thema gemacht!
- Bei der Arbeit mit Scratch habe ich mir die Abfolge der Programmschritte gut vorstellen können!
- Der Zusammenbau der Legoroboter hat mir gut gefallen und ist unserer Gruppe leicht gefallen!
- Bei der Arbeit mit Enchanting konnte ich schnell verstehen, welche Rolle die Motoren und Sensoren eines Roboters für die Programmierung spielen!
- Ich habe schnell einfache Programme selbstständig erarbeiten können.
- Glaubst du, dass dieses Projekt in jeder Informatikgruppe der 4. Klasse gemacht werden sollte?

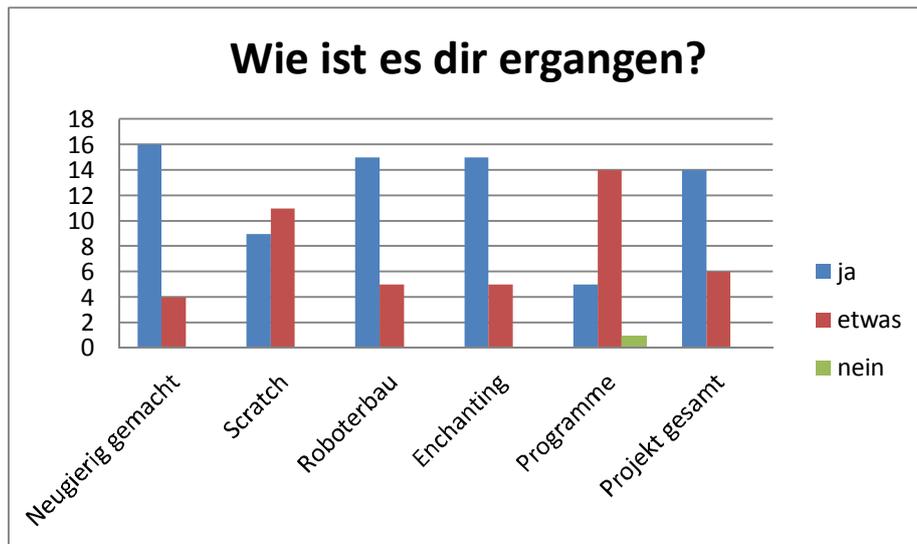


Abbildung 6: Auswertung - Wie ist es dir ergangen?

Es zeigt sich, dass der Lernerfolg beim Thema selbstständige Entwicklung von Programmen nicht so hoch eingeschätzt wurde. Sowohl die Vorstellung der Programmschritte in Scratch wie auch das Schreiben der Programme ist den Schüler_innen schwerer gefallen als der Zusammenbau der Roboter und der Umgang mit Sensoren und Motoren. Nur einmal wurde die Frage, ob die Erarbeitung von Programmen erfolgreich umgesetzt werden konnte, mit „nein“ beantwortet.

Die Selbsteinschätzung zeigt insgesamt eine positive Bewertung des Lernerfolgs. Da die Testergebnisse ebenfalls positiv waren, kann die positive Einschätzung der Schüler_innen als realistisches Bild ihres Erfolgs verstanden werden.

3.1.3 SWOT – Analyse

Die abschließende SWOT-Analyse wurde aus Schüler_innenperspektive durchgeführt. Sie wurden gebeten, zumindest eine Anmerkung pro Feld einzutragen. Uns ist bewusst, dass diese Form der Rückmeldung hohe Anforderungen stellt und die Bereitschaft zur Auseinandersetzung mit dem Geschehen erfordert. Die Schüler_innen haben sich große Mühe gegeben, Antworten zu geben und ließen sich unserer Einschätzung nach dabei von ihren Stimmungen während des Projekts leiten. Wir möchten aus jedem der vier SWOT-Fenster einige Antworten zitieren.

- S – Stärken:** Was hat dir am Projekt – Computational Thinking gefallen?
- W – Schwächen:** Hat dich etwas bei der Arbeit am Projekt gestört? Was hätte besser laufen können?
- O – Chancen:** Was aus diesem Projekt könnte auch in anderen Unterrichtssituationen ähnlich gemacht werden?
- T – Risiken:** Was sollte auf gar keinen Fall so weiter gemacht werden?

S	das Zusammenbauen und Programmieren; Gruppenarbeit; Programmieren mit Scratch und Enchanting; LAOLA-Welle; selbstständiges Programmieren! xd; ...,wenn wer Hilfe brauchte, hat jeder geholfen, wie er konnte, eigentlich alles; das war einmal was Anderes
W	die Lautstärke; Probleme beim Upload auf den Roboter; 2-er Team wären besser; die sich gut ausgekannt haben, waren in einer Gruppe; es hätte Free-style geben sollen; es war alles schwer;..., etwas Anderes aus den Robotern bauen – z.B. einen Greifarm;
O	die Teamarbeit; Gruppen; Gruppenarbeit in Deutsch, Mathe und Englisch; Gruppenarbeit mit Arbeitsteilung; es sollten auch in anderen Gegenständen Roboter gebaut werden;
T	Mehr Arbeit mit dem Roboter, weniger Scratch; nur eine Programmiersprache; mehr Arbeit mit der Software

3.2 Evaluation aus Sicht der Ziele des Themenprogramms

Mit diesem Projekt wurde versucht eine Verbindung von spielerischen Elementen im Informatikunterricht mit einer Einführung in die Grundlagen der Programmierung herzustellen. Vor allem das Interesse für Informatik und technische Fragestellungen konnte durch den Einsatz der Lego-Roboter geweckt werden. Die Schüler_innen gingen mit viel Interesse und Begeisterung an die Thematik des Programmierens heran. Alle Projektteilnehmer_innen, wurden gleichermaßen in das Projekt eingebunden und konnten ihren Stärken und Fähigkeiten entsprechend in den einzelnen Gruppen tätig sein.

Auch der Zusammenhang zwischen Anwender_innenkenntnissen und den dahinterliegenden technischen Grundlagen konnte durch den anschaulichen und schüler_innenzentrierten Unterricht vermittelt werden. So war es für die Schüler_innen kein großes Problem von der handlungsorientierten Ebene (gegenseitiges Programmieren mit Scratchbausteinen) zur abstrakten Ebene des Programmierens der Roboter zu gelangen.

Durch das Arbeiten mit den Lego-Robotern wurden auch Kolleg_innen unsere Schule auf unsere Inhalte im Informatikunterricht aufmerksam und so wurden weitere fächerübergreifende Projekte (Physik, Mathematik und Werken) für das kommende Schuljahr geplant.

3.3 Evaluation aus Sicht übergeordneter IMST Ziele

3.3.1 Gender- und Diversitätsaspekte

Der Grundgedanke des Projektes bestand darin, alle Schüler_innen gleichermaßen in das Arbeiten mit den Robotern bzw. das Programmieren einzubinden. Mädchen, wie auch Buben sollten dabei ihren Fähigkeiten und Interessen entsprechend gefördert werden.

Die sechs Gruppen wurden bewusst geschlechterspezifisch getrennt. Das heißt, es gab fünf „Burschengruppen“ und eine „Mädchengruppe“. Da es insgesamt nur vier Mädchen in der Großgruppe gab, sollten diese auch gemeinsam an dem Projekt arbeiten. Diese geschlechtsspezifische Trennung führte dazu, dass die Mädchen selbstbewusst und eigenverantwortlich ihre Projektthemen bearbeiten konnten. Nur einmal, als es um den Zusammenbau der Roboter ging, ließen sich die Mädchen helfen. Innerhalb der reinen Burschengruppen kam es mehrmals zu Konflikten, da einige Schüler rascher arbeiteten und damit den Lernprozess dominierten.

Da die Ergebnisse des Projektes am Tag der offenen Tür unserer Schule präsentiert wurden, konnten auch weitere Mädchen für das Wahlpflichtfach Informatik im nächsten Schuljahr gewonnen werden.

3.3.2 Schulentwicklungs- und Disseminationsaspekte

Durch dieses Projekt war es möglich den Informatikunterricht so zu gestalten, dass auch andere Lehrer_innen auf das Wahlpflichtfach Informatik aufmerksam wurden. Gemeinsam wurden weitere fächerübergreifende Ideen geboren. Durch den Austausch von Unterrichtsmaterialien konnten auch in anderen Klassen bzw. Unterrichtsgegenständen mit Hilfe des kooperativen Lernens Erfolge erzielt werden. Auch die Lego-Roboter wurden von den Physik – bzw. Mathematiklehrer_innen bzw. wie auch von Professoren der Pädagogischen Hochschule ausgeliehen. So konnten auch angehende Lehrer_innen ihre Programmierkenntnisse durch den Einsatz der Lego Roboter vertiefen.

Insgesamt ermöglichte dieses Projekt eine intensive Vernetzung innerhalb der Schule, wie auch eine schulübergreifende Zusammenarbeit mit der Pädagogischen Hochschule Kärnten und dem BORG Waidmannsdorf.

4 ZUSAMMENFASSUNG

Mit dem Projekt „Computational Thinking – Was läuft im Informatikunterricht?“ gelang es den Schüler_innen der 4. Klasse der NMS das Thema Programmierung näher zu bringen und ihr Interesse zu wecken. Mit Hilfe der Legoroboter arbeiteten die Schüler_innen motiviert und interessiert über Monate hinweg an den Projektinhalten.

Bevor es zur reinen Roboterprogrammierung mit Enchanting kam, wurden die Projektteilnehmer_innen durch handlungsorientiertes und kooperatives Lernen an die jeweiligen Themenbereiche herangeführt. Vor allem der Zusammenbau, Umbau und das direkte Programmieren der Roboter mit Enchanting machte allen viel Spaß.

Da die Struktur jeder Projekteinheiten grundsätzlich gleich aufgebaut war, entwickelten sich der Zugang zum Kooperativen Lernen und die Evaluation am Ende jeder Einheit für die Projektteilnehmer_innen zu einer Selbstverständlichkeit. Aktion, Abstraktion und Reflexion waren die zentralen Kernbereiche, die immer wieder berücksichtigt wurden.

Einzig die Gruppengröße wurde im Laufe der Durchführung verändert. Zuerst arbeiteten die Schüler_innen in Vierer – danach in Zweierteams, somit konnte ein effizienteres und intensiveres Arbeiten ermöglicht werden.

Durch die Anschaffung der Legoroboter können nun auch andere Klassen einen attraktiveren Unterricht erleben. Schüler_innen können mit dieser „neuen Hardware“ begeistert werden und es wird ihnen ein interessanter Zugang zu den MINT-Fächern geboten.

Abschließend ist zu sagen, dass dieses Projekt allen Teilnehmenden sehr viel Spaß gemacht hat. Die Schüler_innen konnten mit Hilfe der Roboterprogrammierung einen wichtigen Bereich der Informatik kennen lernen. Der Informatikunterricht hatte bisher einen Schwerpunkt in der Vermittlung von ECDL – Inhalten. Mit der Einführung der Programmiersprache Scratch wurde Grundwissen über Programmstrukturen und deren Einsatz in der Entwicklung eigener Programme vermittelt. Die Anwendung dieses Wissens bei der Programmierung von Robotern mit Enchanting war insofern sehr spannend, da zusätzliche Komponenten angesprochen werden konnten. Mittels Sensoren einen Roboter zu steuern, erfordert einen weiteren Abstraktionsschritt und erhöht damit die Anforderungen an die Lernenden. Die Möglichkeit unmittelbar auf etwaigen Fehler zu reagieren und durch die veränderte Programmierung zum Erfolg zu gelangen, hielt die Motivation über längere Zeit hoch. Am Tag der offenen Tür wurden die Legoroboter von Schüler_innen selbstständig umgebaut und mittels Handys gesteuert.

Auch die Zusammenarbeit in den einzelnen Gruppen war sehr gut. Die Schüler_innen gingen motiviert an die Arbeit und teilten sich die einzelnen Arbeitsschritte auf. Durch dieses durchwegs positiv belegte Projekt, kamen wir auf die Idee für das nächste Schuljahr ein IMST – Projekt zum Thema „Spielen und Lernen? Logisch!“ einzureichen. Wir hoffen, dass wir die Unterstützung von IMST erhalten und somit unser „Roboterprojekt“ im gesamten nächsten Schuljahr ausbauen und vertiefen können.

5 LITERATUR

BRÜNING, Ludger, SAUM, Tobias (2009). Erfolgreiches Unterrichten durch kooperatives Lernen, 5. Auflage. Essen: Neue Deutsche Schule Verlagsgesellschaft mbH.

LÜCKING, Josef (2009). Lego-Mindstorms Education NXT, Eine Einführung für die Schule, Version Feb.09

RUGE, Wolfgang (2012). Roboter im Film Audiovisuelle Artikulationen des Verhältnisses zwischen Mensch und Technik. Magdeburger Schriftenreihe zur Medienbildung, Band 1. ibidem Verlag, Stuttgart. S. 36

WEBER, Mathias; WIESER, Bernhard (2009). Informatische Konzepte mit Robotern vermitteln - Ein Unterrichtsprojekt für die Sekundarstufe I. In: Gesellschaft für Informatik (Hrsg.): Tagungsband zur GI-Fachtagung "Informatik und Schule 2009 (INFOS2009)". Bonn: Köllen. S. 109-120

Internetquellen:

Legoroboter: URL: http://de.wikipedia.org/wiki/Lego_Mindstorms, [19.03.2013].

Informelles Lernen: URL: http://de.wikipedia.org/wiki/Informelles_Lernen, [19.03.2013].

Referenzhandbuch - Scratch: URL: http://www.olinger.net/iWeb/berufliches/Scratch_files/scratch_referenzhandbuch.pdf, [06.12.2013].

Handbuch – Programmieren für Kinder: <http://www.kinderlabor.ch/cms/upload/Scratch-Unterlagen/Scratch-Handbuch.pdf>, [06.12.2013].

Enchanting Cards: URL: <http://enchanting.robotclub.ab.ca/tiki-index.php>, [20.10.2013].

6 ANHANG

Anhang 1: Thesenblatt

			21.10.2013
Thema:			
Meine Überlegung			
Auswertung			

Lego Roboter - Projekt

Ablauf und Arbeitsweise



INFORMATIK

Was Roboter können?

Computer/Roboter können ...

Arbeitsauftrag

- Überlege alleine was Computer /Roboter alles können.
Defür hast du 2 Minuten Zeit
- Notiere deine wichtigste Überlegung in einem Satz auf deinem Arbeitsblatt!
Defür hast du 2 Minuten Zeit
- Suche in der Klasse möglichst viele Mitschüler und Mitschülerinnen , die deine Ansicht mit ihrer Unterschrift unterstützen. Wer unterschreibt, muss die Überlegung wiederholen können!
Defür hast du 3 Minuten Zeit

Auswertung Computer können ...

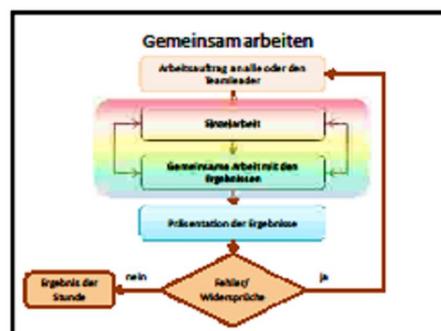
Arbeitsauftrag

- Zähle deine Unterschriften und notiere die Zahl am Arbeitsblatt!
- Die These mit den meisten Unterschriften wird zuerst präsentiert!

**Thesen zum Thema:
was Computer können**

Ergebnispräsentation

These	Anzahl der Unterschriften



Computersprache - Umgangssprache

Was ist ein Algorithmus ?

INFORMATIK

Sprechen Sie technisch

Arbeitsauftrag

Überlege welchen Unterschied du zwischen deiner Umgangssprache und der Programmiersprache Scratch erkennen kannst?

Sprechen Sie technisch

Arbeitsauftrag

Kennst du noch andere Sprachen?
Notiere deine Ideen!

Sprechen Sie technisch

Arbeitsauftrag

Welche Sprache ist wofür am Besten geeignet, ordne zu:

- einen Lösungsweg für Berechnungen aufschreiben Computersprache
- Gefühle ausdrücken Musiknoten Umgangssprache
- einen Liedkomponieren Poetik
- eine logische Abfolge von Anweisungen angeben Mathematik
- Anweisungen zum Backen backen

Algorithmus

Merke

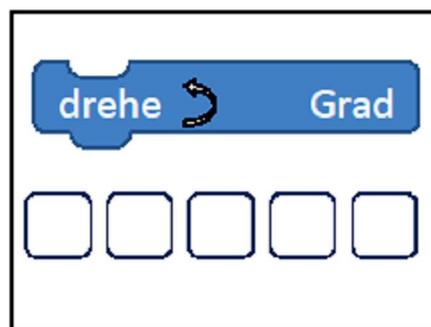
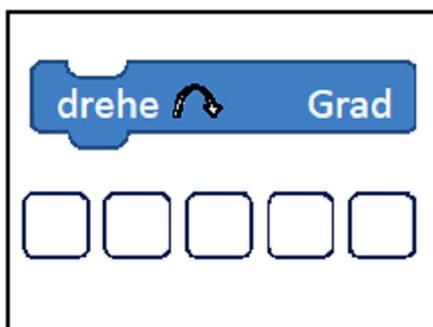
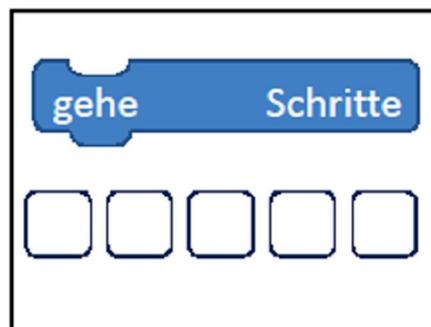
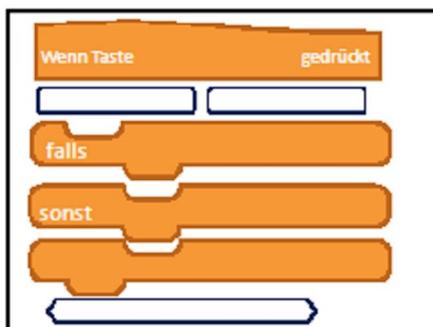
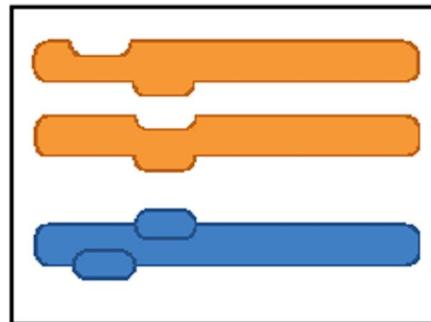
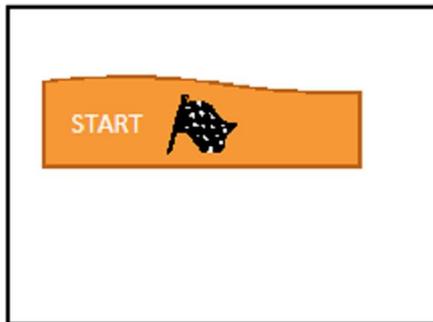
Ein Algorithmus ist eine eindeutige Handlungsvorschrift zur Lösung eines Problems. Algorithmen bestehen aus endlich vielen, wohldefinierten Einzelschritten. Somit können sie zur Ausführung in einem Computerprogramm implementiert werden. Sie können aber auch in menschlicher Sprache formuliert werden. Bei der Problemlösung wird eine bestimmte Eingabe in eine bestimmte Ausgabe überführt.

Erster Algorithmus

Arbeitsauftrag

Der erste für einen Computer gedachte Algorithmus wurde 1843 von Ada Lovelace festgehalten. Sie gilt deshalb als die erste Programmiererin.

Der Algorithmus sollte eine Reihe von Zahlen errechnen und auf der sogenannten Analytical Engine implementiert werden. Weil Charles Babbage seine Analytical Engine nicht vollenden konnte, wurde Ada Lovelaces Algorithmus nie eingesetzt.



Unser erstes Scratch Projekt!



INFORMATIK

Was ist ein Computerprogramm

Arbeitsauftrag

Überlegt einzeln, wie ihr erklären würdet, was ein Computerprogramm ist! 2 min

Einigt euch in der Gruppe auf eine Erklärung! 2 min

Der Präsentator trägt die Gruppenmeinung vor!

Computerprogramm

Merke

Ein **Computerprogramm** oder kurz **Programm** ist eine den Regeln einer bestimmten Programmiersprache genügende **Folge von Anweisungen**, um auf einem Computer eine bestimmte Aufgaben bearbeiten zu können.

Info

Das ist Scratch die Katze!
Sie ist die Akteurin in unseren Programmen.
Zuerst wird's aber du ihre Rolle übernehmen.



Info



Als Akteurin führst du Anweisungen genauso aus, wie sie dir gegeben werden. Du kannst leider keine eigenen Ideen einbringen.
Die Programmierer sind verantwortlich, dass das Programm funktioniert.

Info

Alles beginnt mit dem Start!



Mit Scratch geometrische Figuren zeichnen

Wiederholen von Programmabschnitten - Schleifen

Auftrag

Übertrage folgende Sequenz in den Programmbereich von Scratch! Teste den Ablauf!



Abbildung 4: Eine einfache Sequenz in Scratch

Abschreiben

Unter einer Sequenz versteht man die Abfolge verschiedener Programmschritte die zusammengehören!

Auftrag

Erstelle nun selbstständig Sequenzen! Zeichne Dreiecke, Vierecke und Vielecke!



Abbildung 5: Sequenz zum Zeichnen eines Vierecks

Auftrag

Zeichne nun einen Kreis!
Welchen Winkel solltest du für die Drehung verwenden?

Hinweis: Um einen Block zu kopieren stelle dich an den Beginn des Blocks und klicke die rechte Maustaste. Mit Duplizieren stellst du eine Kopie der Bausteine her!

Abschreiben

Um Programme zu vereinfachen und zu strukturieren, werden sogenannte Schleifen verwendet! Dabei werden einerseits die Programmschritte, die wiederholt werden, angegeben, sowie die Anzahl der Wiederholungen. Die nötigen Steuerungselemente (Bausteine) findest du in der Kategorie Steuerung.

Steuerung des Programmablaufs

Verwendung von Variablen

Abschreiben

Wenn sich während des Ablaufs eines Programms ein vorher festgelegter Wert ändern soll, so verwenden wir dafür eine Variable.

Löse mit Scratch:
Wir wollen mit Hilfe einer variablen Weglänge eine Spirale zeichnen!

Die Figur muss „wissen“, wie lang der Weg war, den sie zuletzt gegangen ist. Bei jedem Programmschritt muss dann der Weg „verlängert“ werden!

Abschreiben

Löse mit Enchanting:
Auf ein akustischen Signal (Los!) soll der NXT zuerst vorwärts, dann ein Stück rückwärts und nochmals vorwärts fahren. Das sollen die NXTs aller Team leicht zeitversetzt machen – LOLA-Welle!
Dann folgt ein zweiter Durchgang mit veränderter Weglänge bzw. Reihenfolge!

Jeder NXT muss „wissen“, wie lang er warten muss bzw. wie weit er vor bzw. zurückfahren soll.

Im Bereich Variablen können Variable definiert werden.

Der aktuelle Wert kann auf der Bühne angezeigt werden!

Der Anfangswert der Variablen wird im setze VARIABLE auf Baustein angegeben. Ebenso kann eine Veränderung festgelegt werden (negative Werte erzeugen eine Verkleinerung des Wertes!)



Auftrag

Öffne Scratch und erzeuge ein beliebiges Objekt. Das Objekt soll in der Mitte der Bühne platziert werden und nach rechts schauen!
Nun soll das Objekt 10 Schritte gehen sich nach unten drehen und 15 Schritte gehen, wieder drehen und 20 Schritte gehen ...




Programm einblenden.

Zusatzaufgabe

Zeichne eine Spirale von außen nach innen, indem du die Weglänge bei jedem Schritt verringerst. Starte dieses Zusatzprogramm mittels Tastendruck (z.B. I).



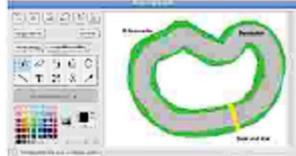

Programm einblenden.

Rennbahn

Programmiere dein erstes Spiel!

Schritt 1

Zeichne im Malprogramm eine Bühne mit einer Rennbahn! Es ist wichtig, dass du einen andersfarbigen Randstreifen zeichnest!



Schritt 2

Programmiere folgende Sequenz: Scratch soll auf den Tastendruck „s“ an der Start/Ziellinie positioniert werden!



Richtung vorgeben x,y Koordinaten Richtung in °

Programm einblenden

Schritt 3

Programmiere folgende Sequenz: Scratch soll nun immer wieder 1 Schritt gehen und überprüfen, ob der grüne Randstreifen berührt wird! Wenn dass der Fall ist wird das Skript gestoppt!



Programm einblenden

Schritt 4

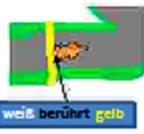
Programmiere folgende Steuerungssequenz: Mit dein Tasten ? Pfeil nach rechts und ? Pfeil nach links soll Scratch eine Richtungsänderung (jeweils um 15°) vornehmen! Trainiere, damit du möglichst schnell durch eine Runde kommst!



Programm einblenden

Schritt 5

Programmiere folgende Steuerungssequenz: Nun muss noch überprüft werden, ob Scratch die Ziellinie überquert! Das ist dann der Fall, wenn die (weiße) Nase die gelben Start/Ziel Markierung berührt!



weiß berührt gelb

Ändere die Maske von Scratch, damit der weiße Bereich groß genug ist!



Programm einblenden

Rechenoperationen mit Scratch



Variable

Variable eignen sich gut um mit ihnen zu rechnen. Scratch unterstützt die Grundrechenoperationen. Die Kacheln befinden sich im Bereich Operatoren.

Um einer Variablen einen bestimmten Wert zuzuordnen wird die Befehlskachel SETZE verwendet.

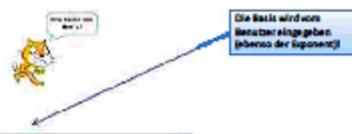
Der Wert kann aber auch vom Spieler (Benutzer) abgefragt werden. Dafür verwendet man die Kacheln FRAGE ...! In der Kachel ANTWORT ist immer der der zuletzt erfragte Wert gespeichert. Dieser kann dann der Variable mit der Kachel SETZE zugewiesen werden.

Projekt: Rechnen mit Scratch Wir wollen a^x errechnen!



Schritt 1

Dafür wählen wir ein beliebiges Objekt? Dieses Objekt fragt den Benutzer welche Potenz berechnet werden soll. Es müssen Basis und Exponent erfragt werden! Wir legen drei Variable an, damit wir mit den eingegebenen Werten aus rechnen können.



Schritt 2

Die Bausteine sind im Bereich Fühlen zu finden!

Schritt 3

Bevor wird die Rechenschritte starten, setzen wird das Ergebnis auf den Wert der Basis (Basis¹ = Basis)! Dann können wir mit der Berechnung beginnen: Die Basis muss nun (n-1)-mal mit sich selbst multipliziert werden. Es gilt $3^4 = 3 \cdot 3 \cdot 3 \cdot 3$! Benutze eine Schleife mit (Exponent - 1) Wiederholungen.



Dafür brauchst du einen Operator

Anhang 10: Enchanting - Cards

Beamer - Einstellungen
(Einstellen wo was einpfedert ist)

Wie können wir Einstellung verändern, dass am NET der Lehrer sein Part 2 angestrichen ist?

1. Wähle auf der Menü
2. Klicke auf
3. Wähle auf der linken Seite den Block mit dem Lichteranschaltungen. Diese blockt ist in
4. Wähle welchen Bereich du verwenden möchtest. Lichter, Performance etc.

5. Wenn du willst kannst du den Namen ändern der Name gibt an

6. Wenn du die Lichter anschaltete der Verfügung

Enchanting

Microkarte

Öffnen der Karte

Wenn NET mit 2 angestrichen ist

Enchanting

Microkarte

Öffnen der Karte

Wenn NET mit 2 angestrichen ist

Enchanting

Führungsmesser

Was will ich messen?

Wenn NET mit 2 angestrichen ist

Enchanting

Führungsmesser

Wie kann ich die Messung steuern?

Wenn NET mit 2 angestrichen ist

Enchanting

Geräuschmesser

Wie kann ich die Geräuschmessung steuern?

Wenn NET mit 2 angestrichen ist

Enchanting

Geräuschmesser

Wie kann ich die Geräuschmessung steuern?

Wenn NET mit 2 angestrichen ist

Enchanting

Zweiradroboter
(Fahre mit deinem Zweiradroboter herum)

Wie kann ich den Roboter steuern?

Wie kann ich die Geschwindigkeit steuern?

Wie kann ich die Richtung steuern?

Wenn NET mit 2 angestrichen ist

Enchanting

Fahren im Quadrat

Wie kann ich den Roboter steuern?

Wenn NET mit 2 angestrichen ist

Enchanting

Fahren im Quadrat

Wie kann ich den Roboter steuern?

Wenn NET mit 2 angestrichen ist

Enchanting

Aufgabe: Fülle die freien Felder aus!

"falls ... sonst" - "Sende START an alle" - "Wenn LEERTASTE gedrückt" - Algorithmus - Anzahl der Wiederholungen - bedingter Anweisungen - Bedingungen - Blöcke - Bühne - Enchanting - endlos - Fahrzeugeinstellungen - festgelegt - Folge von Anweisungen - gestaltet - Kostüme - Motoren - Objekte - Port - Programmiersprache - Schleife - Sensoren - Starthut - übertragen - upload - Variable - verschiedener Programmschritte

Ein Computerprogramm ist eine _____, die den Regeln einer bestimmten _____ genügen. Damit kann auf einem Computer eine bestimmte Aufgaben bearbeitet werden. Eine Handlungsvorschrift, die zur Lösung eines Problems mit Hilfe eines Computers erarbeitet wird, wird auch _____ genannt.

Ein Scratch-Programm beginnt mit dem _____! Die einzelnen Anweisungen stehen als _____ zur Verfügung. Unter einer Sequenz versteht man die Abfolge _____ die zusammengehören! Die _____ dient als Hintergrund für alle _____ eines Programms. Objekte können vom Programmierer selbst _____ werden. Wird ein Objekt in verschiedenen Ausführungen gebraucht, so entwirft man _____.

Programmschritte, die mehrfach wiederholt werden sollen, können in einer _____ zusammengefasst werden. Die _____ kann festgelegt werden. Eine Schleife ohne Wiederholungszahl läuft _____.

Der Ablauf eines Programms kann mittels _____ gesteuert werden. Das Programm muss jederzeit auf diese _____ reagieren.

Ein Beispiel für eine bedingte Anweisung ist: _____. Muss innerhalb einer Sequenz eine Entscheidung getroffen werden, so wird die Anweisung _____ verwendet.

Wenn wir mit mehreren Objekten arbeiten, so können zwischen den Objekten Botschaften gesandt werden, z.B. _____. Wenn sich während des Ablaufs eines Programms ein vorher festgelegter Wert ändern soll, so verwenden wir dafür eine _____.

Um einen Lego-Roboter zu programmieren, kann man die Programmierumgebung von _____ verwenden. Für die Steuerung des Roboters müssen die _____ festgelegt werden. Für die Bewegung sind _____ zuständig, das _____ an dem sie angeschlossen sind, muss _____ werden, ebenso die _____. Bevor ein Programm am Roboter getestet werden kann, muss es _____ werden. Dieser Vorgang heißt _____.

Kannst du dir vorstellen, dass die Art und Weise wie ein Programm geschrieben wird, eine Hilfe bei der Bearbeitung von Themen im Alltag darstellt - finde ein Beispiel bzw. beschreibe eine Situation:

Aufgabe: Fülle die freien Felder aus!

"falls ... sonst" - "Sende START an alle" - "Wenn LEERTASTE gedrückt" - Algorithmus - Anzahl der Wiederholungen - bedingter Anweisungen - Bedingungen - Blöcke - Bühne - Enchanting - endlos - Fahrzeugeinstellungen - festgelegt - Folge von Anweisungen - gestaltet - Kostüme - Motoren - Objekte - Port - Programmiersprache - Schleife - Sensoren - Starthut - übertragen - upload - Variable - verschiedener Programmschritte

Ein Computerprogramm ist eine Folge von Anweisungen, die den Regeln einer bestimmten Programmiersprache genügen. Damit kann auf einem Computer eine bestimmte Aufgaben bearbeitet werden. Eine Handlungsvorschrift, die zur Lösung eines Problems mit Hilfe eines Computers erarbeitet wird, wird auch Algorithmus genannt.

Ein Scratch-Programm beginnt mit dem Starthut ! Die einzelnen Anweisungen stehen als Blöcke zur Verfügung. Unter einer Sequenz versteht man die Abfolge verschiedener Programmschritte die zusammengehören! Die Bühne dient als Hintergrund für alle Objekte eines Programms. Objekte können vom Programmierer selbst gestaltet werden. Wird ein Objekt in verschiedenen Ausführungen gebraucht, so entwirft man Kostüme .

Programmschritte, die mehrfach wiederholt werden sollen, können in einer Schleife zusammengefasst werden. Die Anzahl der Wiederholungen kann festgelegt werden. Eine Schleife ohne Wiederholungszahl läuft endlos .

Der Ablauf eines Programms kann mittels bedingter Anweisungen gesteuert werden. Das Programm muss jederzeit auf diese Bedingungen reagieren.

Ein Beispiel für eine bedingte Anweisung ist: "Wenn LEERTASTE gedrückt" . Muss innerhalb einer Sequenz eine Entscheidung getroffen werden, so wird die Anweisung "falls ... sonst" verwendet.

Wenn wir mit mehreren Objekten arbeiten, so können zwischen den Objekten Botschaften gesandt werden, z.B. "Sende START an alle" . Wenn sich während des Ablaufs eines Programms ein vorher festgelegter Wert ändern soll, so verwenden wir dafür eine Variable .

Um einen Lego-Roboter zu programmieren, kann man die Programmierumgebung von Enchanting verwenden. Für die Steuerung des Roboters müssen die Sensoren festgelegt werden. Für die Bewegung sind Motoren zuständig, das Port an dem sie angeschlossen sind, muss festgelegt werden, ebenso die Fahrzeugeinstellungen . Bevor ein Programm am Roboter getestet werden kann, muss es übertragen werden. Dieser Vorgang heißt upload .

Kannst du dir vorstellen, dass die Art und Weise wie ein Programm geschrieben wird, eine Hilfe bei der Bearbeitung von Themen im Alltag darstellt - finde ein Beispiel bzw. beschreibe eine Situation:

Anhang 12: Arbeitsblatt

Name: _____

In den letzten Informatikstunden hast du mit deinem Team einen Roboter zusammengebaut und programmiert. Vielleicht hast du auch darüber nachgedacht, was Roboter heute und in Zukunft können sollten und was nicht. Bitte schreibe deine Ideen zu den Fragen auf. (Falls der Platz nicht reicht, verwende die Rückseite oder ein Zusatzblatt.)

- ① In deinem Alltag hast du mit zahlreichen technischen Geräten zu tun. Mit den meisten kannst du auch gut umgehen. Hier siehst du Bilder von einigen Geräten. Wo würdest du (d)einen Roboter einordnen! (Markiere den Platz auf der Linie mit einem X)



- ② Begründe deine Entscheidung:

- ③ Wir haben uns gemeinsam den Film *Nummer 5 lebt* angesehen. Welche besonderen Eigenschaften hat Nummer 5 nach dem Blitzschlag? (Schreibe auch auf, warum diese Eigenschaften *besonders* sind.)

- ④ Suche im Internet nach Informationen zu AIBO? Notiere deine Gedanken dazu?

Anhang 13: Evaluation – SWOT Analyse

Anleitung

Wir wollen deine Erfahrungen und Ideen zur Arbeit im IMST-Projekt Computational Thinking erfragen.

Bitte fülle im Blatt Analyse jeden Bereich mit zumindest einigen Stichwörtern aus. Es gibt hier keine richtigen oder falschen Antworten.

Deine Anregungen tragen dazu bei, den Informatikunterricht interessant zu gestalten und den Lernerfolg der Schülerinnen und Schüler zu erhöhen!

Danke für deine Mitarbeit im Projekt und bei dieser abschließenden Befragung!

Für weitere Anmerkungen kannst du gerne die Rückseite verwenden.

Wie ist es dir gungen?	trifft zu:		
	ja	etwas	nein
Das Nachdenken über Computerprogrammierung und Robotik hat mich neugierig auf dieses Thema gemacht			
Bei der Arbeit mit Scratch habe ich mir die Abfolge der Programmschritte gut vorstellen können!			
Der Zusammenbau der Lego-Roboter hat mir gut gefallen und ist an der Gruppe leicht gefallen!			
Bei der Arbeit mit Finch hat ich schnell verstanden, welche Rolle die Motoren und Sensoren eines Roboters für die Programmierung spielen!			
Ich habe schnelle InChica Programme selbstständig erstellen können.			
Glaubst du, dass dieses Projekt in jeder Informa tionsgruppe der 4. Klasse gemacht werden sollte?			

<p>Wie hast du das Projekt (Computational Thinking) geliebt?</p>	<p>Wie hast du das Projekt nicht so geliebt?</p>
<p>Wie hat dir das Projekt am meisten Spaß gemacht?</p>	<p>Was war dir das Projekt am wenigsten gefallen?</p>

Anhang 14: Evaluation – Feedback Bögen

Projekteinheit 1:

Die heutige Einheit hat war:		
		
Meine Teamrolle war:		
Anmerkung:		

Projekteinheit 2:

Was läuft im Informatikunterricht?			
Die heutige Einheit hat war:			
Ich kenne Bewegungsbausteine und kann sie zusammensetzen.			
Meine Teamrolle war:			
Anmerkung:			

Projekteinheit 3:

Was läuft im Informatikunterricht?			
Die heutige Einheit hat war:			
Ich kann Scratch aufrufen und ein Objekt auf einer Bühne bewegen			
Meine Teamrolle war:			
Anmerkung:			

Projekteinheit 4:

Was läuft im Informatikunterricht?			
Die heutige Einheit hat war:			
Ich kann Bühnen und Objekte gestalten!			
Ich kenne die Begriffe: <ul style="list-style-type: none"> ○ Algorithmus ○ Sequenz ○ Schleife und kann sie erklären.			
Meine Teamrolle war:			
Anmerkung:			

Projekteinheit 5:

Was läuft im Informatikunterricht?			
Die heutige Einheit hat war:			
Ich kann ein Objekt mit Hilfe der Pfeiltasten steuern!			
Ich kann eine Bedingung formulieren!			
Meine Teamrolle war:			
Anmerkung:			

Projekteinheit 6:

Was läuft im Informatikunterricht?			
Die Einheit hat war:			
Ich kenne die Bausteine von Enchanting, um Sensoren anzusteuern!			
Ich kann ein Programm auf den Roboter übertragen und starten!			
Meine Teamrolle war:			
Anmerkung:			

Projekteinheit 7:

Was läuft im Informatikunterricht?			
Die Einheit hat war:			
Der Zusammenbau des Roboters hat mir gefallen!			
Ich weiß, worauf ich beim Zusammenbauen achten muss!			
Meine Teamrolle war:			
Anmerkung:			

Projekteinheit 8:

Was läuft im Informatikunterricht?			
Die Einheit hat war:			
Ich kenne die Bausteine von Enchanting, um Sensoren anzusteuern:			
Ich kann ein Programm auf den Roboter übertragen und starten:			
Meine Teamrolle war:			
Anmerkung:			