

MASTERARBEIT | MASTER'S THESIS

Titel | Title

Erforschung eines inklusiven Unterrichtskonzeptes zur Erhöhung der Selbstwirksamkeit im Programmieren von Schüler*innen der 7. Schulstufe, insbesondere von Mädchen, im Informatikunterricht

verfasst von | submitted by

Theresa Schausberger BEd

angestrebter akademischer Grad | in partial fulfilment of the requirements for the degree of
Master of Education (MEd)

Wien | Vienna, 2024

Studienkennzahl lt. Studienblatt |
Degree programme code as it appears on the
student record sheet:

UA 199 500 514 02

Studienrichtung lt. Studienblatt | Degree programme as it appears on the student record sheet:

Masterstudium Lehramt Sek (AB) Unterrichtsfach
Bewegung und Sport Unterrichtsfach Informatik

Betreut von | Supervisor:

Univ.-Prof. Dipl.-Ing. Dr. Renate Motschnig

Abstrakt

Ein Aspekt der existierenden Geschlechterungleichheit im Informatikberufssektor ist die geringere Selbstwirksamkeit im Programmieren der Mädchen. In dieser Masterarbeit wurde der Einfluss eines inklusiven Unterrichtskonzeptes mit den Schwerpunkten auf Game-Based-Learning, Pair-Programming in Scratch, Feedback und Reflexion auf die Selbstwirksamkeit im Programmieren der Schüler*innen der 7. Schulstufe in einem Wiener Gymnasium im Rahmen einer Aktionsforschung untersucht. Durch einen Fragebogen zu der Selbstwirksamkeit im Programmieren ist ein höchst signifikant höherer Wert der Schüler gegenüber den Schülerinnen festgestellt worden. Dieser Wert ist durch die Durchführung des Unterrichtskonzeptes bei den Schülerinnen als auch Schülern höchst signifikant verbessert worden. Dies führte dazu, dass zwischen allen Geschlechtern kein statistischer Unterschied mehr zu erkennen ist. Als größte Limitation in der Aktionsforschung haben sich, die durch die Rahmenbedingungen vorgegebenen, 50-Minuten-Einheiten ergeben und daher wären für weitere Forschung 100-Minuten-Einheiten interessant. Die Arbeit vereint Ansätze der Literatur aus vorheriger Forschung zu einem Gesamtkonzept zur Steigerung der Selbstwirksamkeit im Programmieren, insbesondere der Mädchen, und leitet aus den gewonnenen Ergebnissen Implementierungsstrategien für den Unterricht ab.

Abstract

One aspect of the existing gender inequality in the computer science profession is a lower self-efficacy in programming among girls. In this master's thesis, the influence of an inclusive teaching concept with a focus on game-based learning, pair programming in Scratch, feedback and reflection on the self-efficacy in programming of 7th grade students in a Viennese secondary school was investigated as part of action research. A questionnaire on self-efficacy in programming revealed a significantly higher value for male than for female pupils. This value was improved by the implementation of a teaching concept for both male and female learners so that there is no longer any recognisable statistical difference between genders. The biggest limitation in the action research was the 50-minute units imposed by the framework conditions, which is why 100-minute units would be interesting for further research. The work combines approaches from literature from previous research into an overall concept for increasing self-efficacy in programming, especially for girls, and derives implementation strategies for teaching from the results obtained.

Danksagung

An dieser Stelle möchte ich mich herzlich bei allen bedanken, die mich bei dieser Arbeit unterstützt haben. An erster Stelle meiner Betreuerin Dr. Renate Motschnig, die mir immer mit Rat, konstruktiver Kritik und Hilfe beiseite stand, das ganze Studium ein Vorbild für mich war und eindeutig einen positiven Einfluss auf meine Selbstwirksamkeit als Informatik-Lehrerin hatte. Ebenfalls ein besonderer Dank an meine tollen Schüler*innen, vor allem auch an die Eltern, welche zur Teilnahme an dieser Forschung zugestimmt haben. Außerdem möchte ich mich bei Dr. techn. Dominik Dolezal für die Hilfe beim Erstellen der T-Tests und bei Gregor Steinhuber, MEd für die Intercoderreabilitätsprüfung bedanken. Abschließend gilt ein besonderer Dank meiner Familie (vor allem meinem Vater) und Freunden für den unerschöpflichen emotionalen Rückhalt.

Inhaltsverzeichnis

Abbildungsverzeichnis	8
Tabellenverzeichnis	9
1. Einleitung	10
1.1 Forschungsfragen	11
1.2 Methodisches Vorgehen	11
1.2.1 Auswertung des Fragebogens bezüglich der Selbstwirksamkeit im Programmieren	14
1.2.2 Analyse der offenen Fragen	15
2. Theoretischer Rahmen	19
2.1 Selbstwirksamkeit	19
2.2 Vier Haupteinflussfaktoren der Selbstwirksamkeit	19
2.2.1 Aktive Meistererfahrung	19
2.2.2 Stellvertretende Erfahrung	20
2.2.3 Verbale Überzeugung	20
2.2.4 Physiologische und affektive Zustände	21
2.3 Aktueller Forschungsstand	21
2.3.1 (Geschlechtsspezifische) Selbstwirksamkeit in der Informatik	21
2.3.2 Selbstwirksamkeit in benachbarten Disziplinen	23
2.4 Unterrichtskonzept	24
2.4.1 Lernziele	24
2.4.2 Reflexion/Lerntagebuch	26
2.4.3 Game-Based Learning	28
2.4.4 Pair-Programming	29
3. Planung der Aktionsforschung	31
3.1 Rahmenbedingungen	31
3.2 Einverständniserklärung der Eltern	32
3.3 Fragebogen	32
3.4 Erste Unterrichtseinheit: Programm, Objekt, Schleifen	35
3.5 Zweite Unterrichtseinheit: Wenn-Funktionen	37
3.6 Dritte Unterrichtseinheit: Pair-Programming in Scratch	39
3.7 Vierte Unterrichtseinheit: Pair-Programming in Scratch	42
3.8 Fünfte Unterrichtseinheit: Präsentation und Feedback	43
4. Durchführung und Ergebnisse der Aktionsforschung	45
4.1 Erste Unterrichtseinheit – Schleifen	45

4.1.1	Durchgang 1	45
4.1.2	Durchgang 2	47
4.2	Zweite Unterrichtseinheit – Wenn-Funktionen	48
4.2.1	Durchgang 1	48
4.2.2	Durchgang 2	49
4.3	Dritte Unterrichtseinheit – Pair Programming	49
4.3.1	Durchgang 1	49
4.3.2	Durchgang 2	50
4.4	Vierte Unterrichtseinheit – Pair Programming.....	51
4.4.1	Durchgang 1	51
4.4.2	Durchgang 2	52
4.5	Fünfte Unterrichtseinheit – Präsentation und Feedback.....	53
4.5.1	Durchgang 1	53
4.5.2	Durchgang 2	54
5.	Reflexion und Analyse der Aktionsforschung	56
5.1	Erste Unterrichtseinheit – Schleifen.....	56
5.1.1	Durchgang 1	56
5.1.2	Durchgang 2	57
5.2	Zweite Unterrichtseinheit – Wenn-Funktionen	58
5.2.1	Durchgang 1	58
5.2.2	Durchgang 2	60
5.3	Dritte Unterrichtseinheit – Pair Programming	61
5.3.1	Durchgang 1	62
5.3.2	Durchgang 2	63
5.4	Vierte Unterrichtseinheit – Pair Programming.....	64
5.4.1	Durchgang 1	64
5.4.2	Durchgang 2	65
5.5	Fünfte Unterrichtseinheit – Präsentation und Feedback.....	66
5.5.1	Durchgang 1	67
5.5.2	Durchgang 2	67
6.	Ergebnisse der Fragebögen	69
6.1	Fragebogen 1	69
6.2	Fragebogen 2	70
6.2.1	Offene Freitextfragen.....	72
6.3	Vergleich Fragebogen 1 und 2	75
7.	Diskussion	77

7.1	Limitationen	77
7.2	Diskussion der Ergebnisse der Fragebögen.....	79
7.3	Implikationen für zukünftigen Unterricht	81
8.	Fazit.....	85
9.	Literaturverzeichnis	87
10.	Anhang	90
10.1	Fragebogen zur Selbstwirksamkeit im Programmieren	90
10.2	Unterrichtskonzept während der Forschung	91
10.3	Finales Unterrichtskonzept	95
10.4	Alle Ergebnisse der Fragebögen zur Einschätzung der Selbstwirksamkeit im Programmieren.....	102
10.5	Antworten der offenen Freitextfragen	103
10.6	Elternbrief	108

Abbildungsverzeichnis

Abb. 1 Der Kreislauf von Reflexion und Aktion nach Altrichter und Posch (2007)	12
Abb. 2 Darstellung der Zyklen der Aktionsforschung in dieser Forschungsarbeit.....	13
Abb. 3 Ablaufmodell nach (Mayring & Fenzl, 2014, S. 550)	15
Abb. 4 Benutzungsoberfläche Scratch (abgerufen von https://scratch.mit.edu/projects/editor/?tutorial=getStarted).....	26
Abb. 5 Reflexionskreislauf nach Gibbs 1988 (Esslinger-Hinz & Sliwka, S. 33).....	27
Abb. 6 Übersicht erste Unterrichtseinheit	35
Abb. 7 Lightbot Level 0 (abgerufen von https://lightbot.lu/).....	36
Abb. 8 Übersicht zweite Unterrichtseinheit	37
Abb. 9 Spiel Blockly Level 7 (abgerufen von https://blockly.games/maze?lang=de&level=7&&skin=0)	38
Abb. 10 Lerntagebuch-Angabe auf der Lernplattform Eduvidual	38
Abb. 11 Übersicht dritte Unterrichtseinheit	39
Abb. 12 Beispiel Geschichte von Ada Lovelace und Computer auf Scratch.....	40
Abb. 13 Beispiel zur Erklärung	41
Abb. 14 Aufgabenstellung Pair-Programming.....	41
Abb. 15 Übersicht vierte Unterrichtseinheit.....	42
Abb. 16 Übersicht fünfte Unterrichtseinheit	43
Abb. 17 Übersicht erste Unterrichtseinheit	45
Abb. 18 Übersicht zweite Unterrichtseinheit	48
Abb. 19 Übersicht dritte Unterrichtseinheit	49
Abb. 20 Übersicht vierte Unterrichtseinheit.....	51
Abb. 21 Übersicht fünfte Unterrichtseinheit	53
Abb. 22 Übersicht erste Unterrichtseinheit	56
Abb. 23 Übersicht zweite Unterrichtseinheit	58
Abb. 24 Lerntagebuch aus dem Onlinekurs Klasse B	60
Abb. 25 Übersicht dritte Unterrichtseinheit	61
Abb. 26 Übersicht vierte Unterrichtseinheit.....	64
Abb. 27 Übersicht fünfte Unterrichtseinheit	66
Abb. 28 Fragebogen 1 Gesamtwerte.....	69
Abb. 29 Fragebogen 1 Verteilung der Werte	70
Abb. 30 Fragebogen 2 Verteilung der Werte	71
Abb. 31 Fragebogen 2 Gesamtwerte.....	71

Abb. 32 Auswertung Rolle der Lehrperson	72
Abb. 33 Auswertung Einfluss Selbstwirksamkeit	73
Abb. 34 Auswertung positive Einflussfaktoren	74
Abb. 35 Vergleich Werte Fragebogen 1 & 2	76

Tabellenverzeichnis

Tab. 1 Differenz zwischen ersten und zweiten Fragebogen nach Geschlecht.....	75
--	----

1. Einleitung

Aus meiner bereits eigenen Unterrichtstätigkeit an einem Wiener Gymnasium konnte ich in dem schulautonomen Wahlpflichtfach „Einführung in die Informatik“ in der siebten und achten Schulstufe beobachten, wie anfänglich vor allem die Schülerinnen besonders im Programmieren keinen beziehungsweise wenig Glauben an sich selbst und ihre Fähigkeiten haben. Umso erfreulicher ist es zu sehen, wie diese Beobachtung im Laufe des Schuljahres verschwindet und man keinen offensichtlichen Unterschied zwischen den Geschlechtern mehr erkennen kann. Für mich stellt sich dadurch die Frage, ob es an mir als Lehrperson in der Rolle als weibliches Vorbild liegt oder doch andere Faktoren ausschlaggebender sind.

Da es in der Literatur nachgewiesen ist, dass Selbstwirksamkeit viel bezüglich der Leistung bewirken kann, stellen Wege zur Erhöhung der Selbstwirksamkeit im Informatikunterricht ein attraktives Forschungsgebiet dar.

Die Selbstwirksamkeit bewirkt nach Bandura (1978), dass Personen mit hoher Selbstwirksamkeit schwierige Aufgaben mit mehr Zuversicht meistern und dadurch für Rückschläge resilienter sind. Nach Banduras (1978) Theorie der Selbstwirksamkeit, kann man diese durch erreichte Leistungen, stellvertretende Erfahrung, verbale Überredung und psychologische Zustände (wie weniger Angst zu haben) erhöhen.

Bereits Tellhed et al. (2022) untersuchten den Unterschied in der Programmier-Selbstwirksamkeit zwischen den Geschlechtern bei 12-jährigen Schüler*innen in Schweden. Sie stellten fest, dass die Jungen eine höhere Selbstwirksamkeit aufwiesen, die auf die Quellen der Selbstwirksamkeit Banduras (1977) zurückzuführen war. Die Jungen hatten mehr Meistererfahrungen mit Computern, mehr Konfrontationen mit einflussreichen Vorbildern und erhielten mehr Ermutigung und Lob für ihre Fähigkeiten im Umgang mit Computern als Mädchen. Eine weitere Studie von Allaire-Duquette et al. (2022) zeigte, dass schon zwei Stunden ausreichen können, um die Selbstwirksamkeit von Mädchen in Bezug auf das Programmieren zu erhöhen. Dies wurde in einem zweistündigen Museum-Workshop belegt. Durch die erfolgreichen Programmiererfahrungen und eine sichere und unterstützende Umgebung konnten die geschlechtsspezifischen Unterschiede in der Selbstwirksamkeit reduziert oder sogar eliminiert werden.

Das Ziel der Arbeit besteht zunächst darin, herauszufinden, ob geschlechtsspezifische Unterschiede in der Selbstwirksamkeit bezüglich des

Programmierens bei Schüler*innen in der 7. Schulstufe eines Wiener Gymnasiums im 16. Bezirk existieren. Des Weiteren wurde ein spezifisches Unterrichtskonzept erstellt und anschließend evaluiert, ob dieses eine Steigerung der Selbstwirksamkeit im Programmieren bewirken kann und welche Faktoren vor allem dazu beigetragen haben. Die Arbeit soll einen Beitrag zur Forschung im Bereich der Programmier-Selbstwirksamkeit, insbesondere von Mädchen, leisten, um der existierenden Genderkluft im Informatiksektor entgegenzuwirken. Ebenso sollen gleichzeitig praktische Implikationen für den Informatikunterricht für andere Lehrer*innen oder Bildungsinstitutionen geboten werden. Aus diesen Zielen wurden folgende Forschungsfragen abgeleitet.

1.1 Forschungsfragen

Existieren an einem Wiener Gymnasium in der 7. Schulstufe geschlechtsspezifische Unterschiede bezüglich der Selbstwirksamkeit im Programmieren vor der Durchführung des Unterrichtskonzeptes?

Zeigt ein spezifisches Unterrichtskonzept mit verschiedenen Unterrichtsdessigns und Ansätzen eine Steigerung der Selbstwirksamkeit im Programmieren bei Schüler*innen der 7. Schulstufe, insbesondere von Mädchen? Kann man mit diesem Unterrichtskonzept die möglichen geschlechtsspezifischen Unterschiede minimieren?

Welche Faktoren nennen Schüler*innen als jene, die ihr Vertrauen in ihre Fähigkeiten beim Programmieren beeinflussen?

Welche didaktischen Implementierungsstrategien für den Unterricht können aus den Forschungsergebnissen abgeleitet werden, um im schulischen Umfeld genderspezifische Unterschiede in Bezug auf Gender in der Informatik zu reduzieren und somit effektiv der existierenden Genderkluft im Informatikberufssektor entgegenzuwirken?

1.2 Methodisches Vorgehen

Grundsätzlich beruht diese Arbeit auf der Aktionsforschung nach Altrichter und Posch (2007).

„Aktionsforschung ist die systematische Untersuchung beruflicher Situationen, die von Lehrerinnen und Lehrern selbst

durchgeführt wird, in der Absicht, diese zu verbessern (Elliott, 1981, zitiert nach Altrichter und Posch, 2007, S.13).“

Unter Aktionsforschung wird laut dieser Definition die Erforschung des eignen Unterrichts, um diesen zu verbessern, verstanden. In dieser Arbeit geht es grundlegend darum, herauszufinden, wie die Selbstwirksamkeit im Programmieren von Schüler*innen, insbesondere von Mädchen, mit einem bestimmten inklusiven Unterrichtskonzept erhöht werden kann. Die Aktionsforschung nach Altrichter und Posch (2007) unterliegt folgenden Zyklen, welcher in Aktion und Reflexion unterteilt wird.

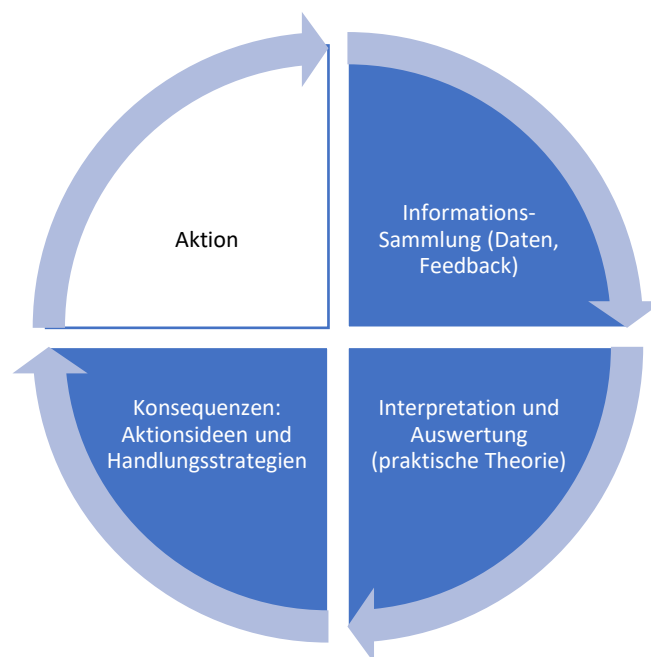


Abb. 1 Der Kreislauf von Reflexion und Aktion nach Altrichter und Posch (2007)

In der Abbildung 1 stellt das weiß gefärbte Feld die Aktion im Unterricht dar. Die blau gefärbten Felder spiegeln die Phase der Reflexion wider, bestehend aus Informationssammlung, Interpretation und Auswertung sowie Konsequenzen daraus.

Bevor mit der Aktionsforschung begonnen werden kann, gehört ein Einstieg nach Altrichter und Posch (2007) geplant. Der Schwerpunkt dieses Abschnittes ist die Entwicklung eines Ausgangspunktes für die eigene Forschungstätigkeit. In dieser Arbeit beruht der Ausgangspunkt auf der Lehrperson und ihre Lehrtätigkeit in den Klassen. Genauer zum Einstieg ist in dem Kapitel „Planung der Aktionsforschung“ nachzulesen.

In der Phase der Aktion werden „Handlungen in komplexen Situationen ... gesetzt (manchmal als Umsetzung von ‚Plänen‘, manchmal spontan) (Altrichter & Posch, 2007, S. 27).“ Gefolgt wird diese Phase von der Datensammlung. Hierzu gehören alle Sammlungen von Daten, Erfahrungen, Dokumenten und weiteres, welche in der Phase der Aktion entstanden sind. In dieser Forschungsarbeit werden vor allem Beobachtungen der Lehrperson und die Reflexionen in den Lerntagebüchern der Schüler*innen gesammelt. Diese Daten werden anschließend interpretiert und nach den Zielen dieser Arbeit ausgewertet. Das Ziel ist die Steigerung der Selbstwirksamkeit im Programmieren, insbesondere von Mädchen. Aus dieser Auswertung und Interpretation lassen sich anschließend Konsequenzen ableiten, welche die Unterrichtseinheit in der anderen Klasse verändern. Anschließend beginnt der Zyklus wieder von vorne mit der Aktion und durchläuft alle Phasen ein weiteres Mal. Nachdem wieder Konsequenzen abgeleitet wurden, steht die schlussendliche Unterrichtseinheit fest und wird zum finalen Gesamtkonzept in dieser Form hinzugefügt. Dieses Unterrichtskonzept mit den gewonnenen Ideen und Erfahrungen wird anschließend im Sinne von Altrichter und Posch (2007) in Form dieser Masterarbeit verbreitet.

Aufgrund der Rahmenbedingungen, welche im Kapitel „Planung der Aktionsforschung: Rahmenbedingungen“ noch genau beschrieben werden, konnte nicht das gesamte Konzept auf einmal durchgeführt und anschließend wiederholt werden. Stattdessen ergaben sich kleine eigene Zyklen innerhalb einer Unterrichtseinheit, welche in Abbildung 2 genauer dargestellt werden.

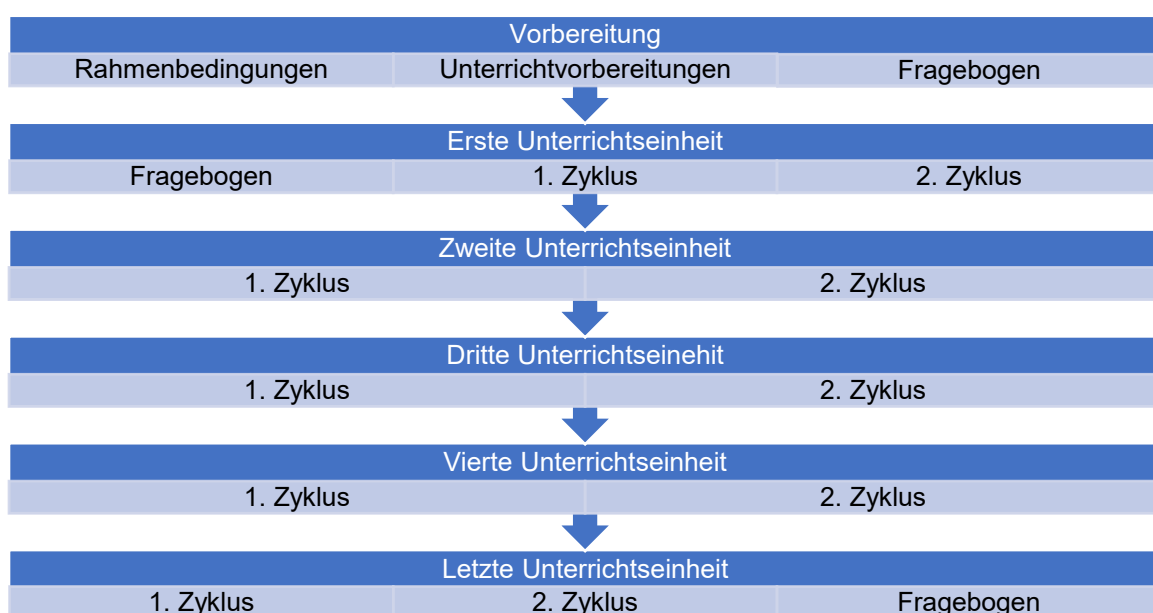


Abb. 2 Darstellung der Zyklen der Aktionsforschung in dieser Forschungsarbeit

Somit startete die Forschung mit einer Analyse der Rahmenbedingungen und einer ausführlichen Literaturrecherche, welche im Kapitel 2 „Theoretischer Rahmen“ zu finden ist. Anschließend begannen die Vorbereitungen für die Forschung, welche aus dem der Literaturrecherche aufbauenden groben Unterrichtskonzept sowie der Erstellung des Fragebogens bestand. Der Fragebogen wurde am Anfang und am Ende der Forschung mit den Schüler*innen erhoben, um Veränderungen in der Selbstwirksamkeitserwartung feststellen zu können. Genaue Details zu den Fragebögen sind im Kapitel „3.3 Planung der Aktionsforschung – Fragebogen“ angeführt. Grundsätzlich bestand der Fragebogen aus zehn Aussagen, die darauf abzielten, die Selbstwirksamkeit im Programmieren festzustellen. Die Schüler*innen konnten sich hierbei auf einer vierteiligen Skala zwischen „gar nicht wahr“ und „vollkommen wahr“ einschätzen. Nach Ausfüllen des ersten Fragebogens begann die eigentliche Aktionsforschung mit der ersten Unterrichtseinheit, welche in einer Klasse durchgeführt worden ist (Aktion-Phase). Anschließend wurde sie reflektiert, sowie verbessert und anschließend wieder in Aktion in der Klasse B gesetzt. Nach der Aktion wurde der Zyklus wieder vollendet, konnte jedoch aufgrund der Rahmenbedingungen kein drittes Mal mehr in Aktion treten. Die gleichen Zyklen wiederholten sich in den Unterrichtseinheiten zwei bis fünf. Zum Abschluss der Aktionsforschung bekamen die Schüler*innen wieder den gleichen Fragebogen mit zwei zusätzlichen offenen Fragen.

Inwiefern spielt für dich die Lehrperson eine Rolle bei der Begleitung beim Programmieren-Lernen?

Hast du während der letzten Unterrichtsstunden eine Veränderung deines Vertrauens in deine Fähigkeiten beim Programmieren bemerkt? Wenn ja, beschreibe welche Faktoren dazu beigetragen haben?

1.2.1 Auswertung des Fragebogens bezüglich der Selbstwirksamkeit im Programmieren

Die Schüler*innen erreichten je nach Einschätzung bei einer Aussage zwischen einen („gar nicht wahr“) bis zu vier Punkten („vollkommen wahr“). Am Ende wurde die durchschnittliche Punkteanzahl aller zehn Aussagen berechnet. Da die Schüler*innen bei den Fragebögen ihren Namen vermerken mussten, konnte anschließend zusätzlich zu den einzelnen Scores ebenso die Entwicklung im Punktesystem betrachtet werden. Um die Signifikanz der Ergebnisse festzustellen, wurden insgesamt vier T-Tests durchgeführt.

1.2.2 Analyse der offenen Fragen

Die offenen Fragen wurden mittels der qualitativen Inhaltsanalyse nach Mayring ausgewertet. Das Ziel ist die folgenden Forschungsfragen noch genauer beantworten zu können:

Zeigt ein spezifisches Unterrichtskonzept mit verschiedenen Unterrichtsd designs und Ansätzen eine Steigerung der Selbstwirksamkeit im Programmieren bei Schüler*innen der 7. Schulstufe, insbesondere von Mädchen?

Welche Faktoren nennen Schüler*innen als jene, die ihr Vertrauen in ihre Fähigkeiten beim Programmieren beeinflussen?

Zur Auswertung des Fragebogens wurde die qualitative Inhaltsanalyse nach Mayring verwendet. Die qualitative Inhaltsanalyse erfolgte nach Mayring und Fenzl (2014) einem bestimmten Ablaufmodell mit induktiver Kategorienbildung und deduktiver Kategorienanwendung.

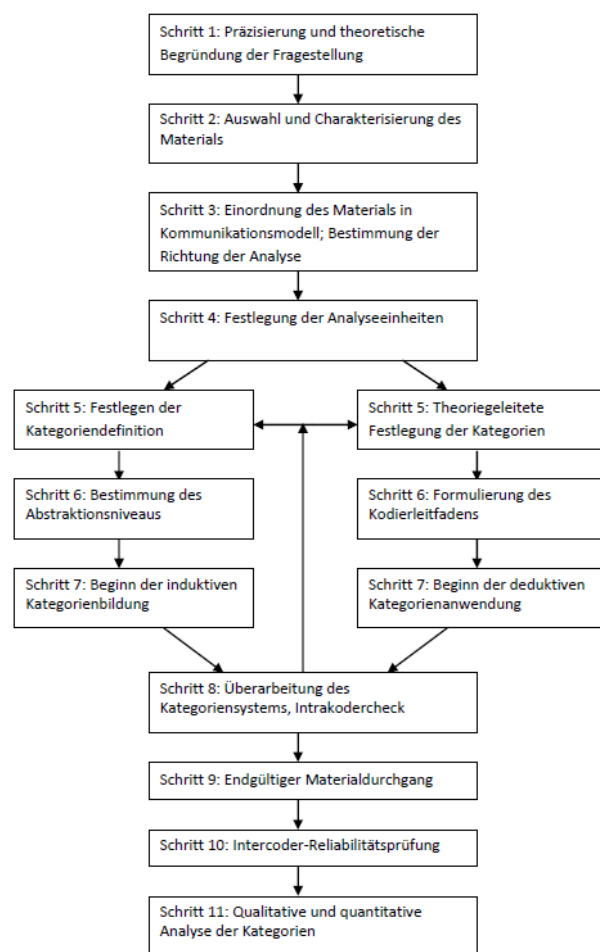


Abb. 3 Ablaufmodell nach (Mayring & Fenzl, 2014, S. 550)

In dieser Forschung wurde sowohl die induktive Kategorienbildung als auch die deduktive Kategorienanwendung verwendet. Die Präzisierung und theoretische Begründung der Fragestellung aus Schritt 1 kann in den Kapiteln „Einleitung“ als auch „Theoretischer Rahmen“ nachgelesen werden.

Zur Auswahl und Charakterisierung des Materials wurden die Antworten der offenen Fragen des zweiten Fragebogens ausgewertet. Die Fragen wurden online auf der Lernplattform Eduvidual von den Schüler*innen ausgefüllt.

Bezüglich Einordnung des Materials in ein Kommunikationsmodell und Bestimmung der Richtung der Analyse sind die Textproduzenten somit Schüler*innen aus der 7. Schulstufe einer AHS im 16. Bezirk, Wien. Im Rahmen des Informatik-Unterrichtes wurden die Texte erstellt. Die Zielgruppe sind Informatik-Lehrperson, sowie auch Informatik-Lehramt Studierende, die Informatik mit dem zusätzlichen Benefit der Steigerung der Selbstwirksamkeit im Programmieren unterrichten wollen. Richtung der Analyse war, eine mögliche Steigerung der Selbstwirksamkeit im Programmieren sowie die Faktoren dazu festzustellen, als auch die Rolle der Lehrperson in diesem Kontext.

Die Analyseeinheiten, eine Antwort einer*s Schüler*in wurden spezifisch folgendermaßen festgelegt: Eine Kodiereinheit, der kleinste auszuwertende Materialbestandteil, entspricht ein Wort einer Antwort. Die Kontexteinheit ist eine ganze Beantwortung einer Frage einer*eines Schüler*in. Alle Antworten aller Schüler*innen bilden die Auswertungseinheit.

Die Kategorien wurden grundsätzlich induktiv gebildet. Bei der ersten Frage zu der Steigerung der Selbstwirksamkeit im Programmieren sowie der Faktoren wurden Kategorien auch zusätzlich im Vorhinein deduktiv gebildet. Dies waren die vier großen Einflussfaktoren auf die Selbstwirksamkeit nach Bandura (1997), sowie positive, gleichbleibende und negative Steigerung der Selbstwirksamkeit. Nach den vier Haupteinflussfaktoren nach Bandura (1997) sind die Kategorien aktive Meistererfahrung, stellvertretende Erfahrung, verbale Überzeugung und physiologische/affektive Zustände. Bei der zweiten offenen Frage wurden die Kategorien Vorbild sowie kein Vorbild deduktiv gebildet.

Induktiv ergaben sich nach nochmaliger Überarbeitung nach dem Intrakodercheck noch 15 weitere Kategorien. Somit entwickelte sich folgendes Kategoriensystem. Die ersten sieben Kategorien wurden für die erste offene Frage

„Inwiefern spielt für dich die Lehrperson eine Rolle bei der Begleitung beim Programmieren-Lernen?“ verwendet. Für die zweite offene Frage im Fragebogen entstanden die Kategorien 8 bis 10.6. Somit waren die Kategorien bezüglich der ersten Frage:

- 1 Feedbackfunktion
- 2 wichtige/große Rolle
- 3 neue Lehrinhalte vermitteln
- 4 keine Rolle
- 5 Lehrperson als Hilfe bei Problemen/Fragen
- 6 kein Vorbild
- 7 Vorbild

Die restlichen Kategorien betreffend der zweiten Frage bildeten:

- 8 kein Einfluss Selbstwirksamkeit
 - 8.1 Verpasster Unterricht
 - 8.2 eigene Vorerfahrung
- 9 negativer Einfluss Selbstwirksamkeit
 - 9.1 Überforderung
- 10 positiver Einfluss Selbstwirksamkeit
 - 10.1 Physiologische, affektive Zustände
 - 10.2 Verbale Überzeugung
 - 10.2.1 Verbale Überzeugung der Lehrperson
 - 10.3 stellvertretende Erfahrung
 - 10.4 Aktive Meistererfahrung
 - 10.4.1 Erfahrung Pair-Programming Scratch
 - 10.5 Motivation
 - 10.6 Verbesserung der eigenen Fähigkeit

Anschließend kam es zum endgültigen Materialdurchgang und die Antworten der beiden Fragen wurden dementsprechend codiert.

Für Intercoder-Übereinstimmung stellte sich Gregor Steinhuber, MEd zur Verfügung und codierte die Antworten unabhängig noch einmal. Bei der Variante 1 der Intercoder-Übereinstimmung konnten 100% erreicht werden. Hier wird überprüft, ob jeder Code des ersten Durchgangs auch von dem zweiten Codierer angewandt worden ist.

Der elfte und letzte Schritt war die qualitative und quantitative Analyse der Kategorien und wird im Kapitel „Ergebnisse“ sowie „Diskussion“ dargestellt.

2. Theoretischer Rahmen

2.1 Selbstwirksamkeit

Bandura (1997) beschreibt sein Buch, welches den Namen „Self efficacy – the exercise of control“ trägt, folgendermaßen:

„This book explores the exercise of human agency through people's beliefs in their capabilities to produce desired effect by their actions. It reviews in considerable detail the origins of efficacy beliefs, their structure, the processes through which they affect human well-being and accomplishments, and how these processes can be developed and enlisted for human betterment“ (Bandura, 1997, S. 7)

Diese Beschreibung entspricht dem Konzept der Selbstwirksamkeit. Grundsätzlich kann man sagen, dass Selbstwirksamkeit die Überzeugung widerspiegelt, dass man den Glauben an sich selbst hat, eine bestimmte Handlung auszuführen und dadurch zu einem gewünschten Ziel kommt, laut Bandura (1978). Des Weiteren betont Bandura (1978), dass die Selbstwirksamkeitserwartung ein wichtiger Faktor in der Verhaltenspsychologie ist und sich auf verschiedene Bereiche unseres Lebens auswirkt, wie zum Beispiel auf die Bildung, aber auch Arbeit, Sport und vieles mehr. Dies führt ferner dazu, dass Menschen mit einer hohen Selbstwirkungserwartung Herausforderungen besser bewältigen können und somit ihre Ziele erreichen. Im Gegensatz dazu haben Menschen mit niedriger Selbstwirksamkeitserwartung eine geringere Anstrengungsbereitschaft und dies kann zu minderwertigerer Leistung führen, wie Bandura (1978) erklärt.

2.2 Vier Haupteinflussfaktoren der Selbstwirksamkeit

Nach Bandura (1997) gibt es vier sogenannte „Quellen“ der Selbstwirksamkeit. Dazu zählt die stellvertretende Erfahrung, verbale Überredung, physiologische und affektive Zustände und die aktive Meistererfahrung beziehungsweise auch genannt „eigene Erfahrung“ (Kriegler-Kastelic, 2018, S. 3).

2.2.1 Aktive Meistererfahrung

„Enactive mastery experiences are the most influential source of efficacy information because they provide the most authentic evidence of whatever one can muster whatever it takes to succeed (Bandura, 1997, S. 80).“ Laut Bandura (1997) ist

es von Bedeutung, dass Erfolge nicht ausschließlich auf einfache Faktoren zurückzuführen sind, da dies zu einem leichtfertigen Aufgeben im Falle von Versagen führen kann. Der entscheidende Faktor ist das Erlernen der Fähigkeit, durch ausreichende Anstrengungen das gewünschte Ziel zu erreichen. In weitere Folge führt dies laut Bandura et al. (1980) dazu, dass Personen mit hoher Selbstwirksamkeit Misserfolge mit veränderbaren Faktoren verbinden und somit schwieriger zu einer Enttäuschung führen. Kriegler-Kastelic (2018) begründet ähnlich, dass Menschen mit hoher Selbstwirksamkeit sich Szenarien in der Zukunft positiv beziehungsweise erfolgreich vorstellen und dadurch zusätzliche Anhaltspunkte gewinnen, welche schlussendlich dazu führen, dass sie die Situationen erfolgreich beenden können.

2.2.2 Stellvertretende Erfahrung

Kriegler-Kastelic (2018) hebt hervor, dass abgesehen von der eigenen erfolgreichen Erfahrung, ebenso auch das Beobachten erfolgreichen Handelns als wirksam erwiesen hat. Umso ähnlicher die Person ist, welche die Handlung ausführt, umso stärker kann dadurch die Selbstwirksamkeit beeinflusst werden. Bandura (1997) begründet dies damit, dass Menschen sich nicht nur auf ihre eigene Erfahrung verlassen, sondern weitere Informationsquellen suchen. Aufgrund der fehlenden objektiven Kriterien zur Messung der Durchführbarkeit von verschiedenen Tätigkeiten, orientieren sich Menschen an den Leistungen anderer, um ihre eigenen Fähigkeiten spezifischer einschätzen zu können.

Laut Bandura (1997) spielt das Beobachtungslernen eine entscheidende Rolle, da die Beobachtung einer erfolgreichen Handlungsausführung einer anderen Person die Selbstwirksamkeit des Beobachters erhöhen kann. Stone (2017) fasst die vier Bedingungen des Beobachtungslernens nach Bandura (1997) in die vier Bedingungen Aufmerksamkeit (auf das Modell), Erinnerung (an das beobachtete Verhalten), Reproduktion (physisch und mental) und Motivation (für das Nachahmen des Verhalten) zusammen.

2.2.3 Verbale Überzeugung

Mündliche Überredung oder auch positives Feedback kann helfen, die Selbstwirksamkeit zu steigern. „Die größte Wirkung wird erzielt, wenn der/die Unterstützende als besonders glaub- und vertrauenswürdig wahrgenommen wird und über ExpertInnenwissen verfügt (Kriegler-Kastelic, 2018, S. 6).“ Bandura (1997)

hingegen bekräftigt, dass das Feedback die persönlichen Fähigkeiten hervorheben muss oder dass die Verbesserung der Fähigkeit auf die Anstrengungen zurückzuführen sind, die die Selbstwirksamkeit besonders erhöhen. Weniger stark prägt das Feedback, dass sie aufgrund der Fortschritte für die Aufgabe geeignet sind. Jedoch kann die verbale Überzeugung auch das Gegenteil bewirken, wenn sie als unglaublich eingeschätzt wird. Dadurch wird sie als demotivierend wahrgenommen und somit schwächt sie nach Kriegler-Kastelic (2018) die Selbstwirksamkeit.

2.2.4 Physiologische und affektive Zustände

Nach Bandura (1997) deuten Menschen ihre körperliche Aktivierung als Symbol für eine Anfälligkeit für Dysfunktionen und durch diese hohe Erregung kann die Leistung beeinträchtigt werden. Dies bedeutet wiederum, dass Menschen eine höhere Selbstwirksamkeitserwartung haben, wenn sie nicht dieser Stressreaktion ausgesetzt sind. Die Überwindung emotionaler Reaktionen, wie von Bandura (1997) beschrieben, erhöht den Glauben an die eigene Bewältigungsfähigkeit und kann somit die Leistung verbessern. Laut Kriegler-Kastelic (2018) sind körperliche Reaktionen wie feuchte Hände, Zittern oder Herzrasen ein Indikator für erhöhte körperliche Aktivierung. Durch gezielte Übungen kann eine alternative Interpretation dieser Symptome gefunden werden, indem die Situation akzeptiert und die Vorteile, wie gesteigerte Aufmerksamkeit, erkannt werden.

2.3 Aktueller Forschungsstand

2.3.1 (Geschlechtsspezifische) Selbstwirksamkeit in der Informatik

Allgemein fanden Morales-Navarro et al. (2023) das die Selbstwirksamkeitsüberzeugungen der Schüler*innen im Bereich des Lernens in der Informatik signifikant mit dem „growth mindset“ zusammenhängt. Mit diesem Mindset ist die Überzeugung gemeint, dass Intelligenz und Fähigkeiten durch Lernen, Anstrengung und Beharrlichkeit entwickelt werden können. Die Forschungsergebnisse zeigen, dass die Selbstwirksamkeitsüberzeugung und positive Denkweisen, vor allem das „growth mindset“ eine entscheidende Rolle bezüglich ihres Engagements und Erfolg im Lernen der Informatik spielen.

Die Studie von Tellhed et al. (2022) stellte Geschlechterunterschiede bei 373 Kindern im Alter von zwölf Jahren bezüglich der Selbstwirksamkeit beim Programmieren in Schweden dar. Jungen wiesen zu allen drei Zeitpunkten der

Datenerhebung, zu Schuljahresbeginn, nach der Programmierausbildung, sowie am Schuljahresende eine höhere Programmier-Selbstwirksamkeit als die Mädchen auf. Die Ergebnisse von Tellhed et al. (2022) deuten darauf hin, dass die Selbstwirksamkeit beim Programmieren und die diesbezüglich entstandenen Geschlechtsunterschiede mit den Faktoren, die vorherige Meistererfahrungen, verbale Überzeugungen, beobachtete Erfahrung und Vorbilder zusammenhängen. Letztendlich hatten die Jungen davor mehr Erfahrungen im Umgang mit Computern (Meistererfahrungen). Sie erhielten mehr Lob und Ermutigungen (soziale Überzeugungen), sowie hatten sie nach der Vermutung der Autor*innen mehr Vorbilder in diesen Zusammenhang. Dies alles führte laut Tellhed et al. (2022) zu der höher festgestellten Selbstwirksamkeit. Des Weiteren betonen Tellhed et al. (2022) die Wichtigkeit, dass in der Schule alle Geschlechter die Möglichkeit haben sollen, genau diese Faktoren zu erfahren, um somit auch das Interesse von Mädchen an der Programmierung zu fördern und darüber hinaus Anstrengungen unternommen werden, um negativen Stereotypen entgegenzuwirken.

Zu ähnlichen Ergebnissen kamen Allaire-Duquette et al. (2022) bei ihrer Studie zu einem Museum-Workshop und dessen Veränderungen in der Selbstwirksamkeit. Vor dem Workshop gaben die Mädchen eine allgemein geringere Selbstwirksamkeit bezüglich des Programmierens als die Jungen an. In dem Workshop lernten die Schüler*innen die Programmierung mit Lego Mindstorms. Genau wie Tellhed et al. (2022) führen die Forscher*innen die Entwicklung der Selbstwirksamkeit auf die Meistererfahrungen zurück, welche sie bei diesem Workshop sammeln konnten. Ein weiterer erforschter Faktor ist die Betonung von Allaire-Duquette et al. (2022) der nicht-formalen Lernumgebungen, wie in diesen Fall das Museum, genauso wie die Bedeutung der sicheren und unterstützenden Umgebung. Dies ist konsistent mit der sozialen Überzeugung von Tellhed et al. (2022).

Wie schon Tellhed et al. (2022) und Allaire-Duquette et al. (2022) in den oben beschriebenen Studien höhere Selbstwirksamkeitswerte von Jungen als die der Mädchen in der Informatik verifizieren, kam auch Sullivan (2015) in seiner Studie, bei der 164 Schüler*innen zwischen 12 und 17 Jahren befragt wurden. Die Geschlechterungleichheit beeinflusst die Wahrscheinlichkeit von Frauen, ein Informatikstudium an der Universität zu absolvieren im Vergleich zu Männern. Mädchen betrachten es ebenfalls als weniger wahrscheinlich ein Informatikstudium überhaupt absolvieren zu können. Jedoch konnte in der Studie auch erforscht werden,

dass die Teilnahme von Schülerinnen an einem Programm namens „CodePlus“ zu einer Steigerung der Selbstwirksamkeit führte. Bei diesem Programm nahmen die Schülerinnen an wöchentlichen, zweistündigen Workshops teil, bei dem sie sich mit Themen wie algorithmischen Denken, gesellschaftlichen Aspekten und Programmierung mit Scratch in Gruppenarbeiten auseinandersetzten.

2.3.2 Selbstwirksamkeit in benachbarten Disziplinen

In der Studie von Weinhardt (2017) wurden Schüler*innen der fünften, neunten und zwölften Schulstufe befragt, wie sie ihre Fähigkeiten in Deutsch und Mathematik einschätzen und welche tatsächlichen Noten sie in den Fächern erzielten. In der fünften Schulstufe beginnen die Schülerinnen ihre Fähigkeiten in Mathematik zu unterschätzen und dies hält sich bis in die zwölfte Schulstufe. Dies führt laut dem Autor*in dazu, dass Frauen MINT-Fächer seltener studieren und der Frauenanteil in MINT-Beruf niedrig bleibt. Dies führt Weinhardt (2017) auf Rollenbilder als auch den Einfluss von Mitschüler*innen, Eltern und Lehrpersonen zurück. Ebenso erwähnt Weinhardt (2017) den Effekt, dass die Schülerinnen in der Grundschule im Vergleich zu Jungen über bessere verbale Fähigkeiten verfügen. Eine mögliche Konsequenz davon könnte nach Weinhardt (2017) sein, dass die Mädchen ihre mathematischen Fähigkeiten im Vergleich zu den sprachlichen geringer einschätzen. Deshalb empfiehlt Weinhardt (2017), dass in der Grundschule Maßnahmen ergriffen werden sollten, um die Selbstwirksamkeit der Mädchen in ihren mathematischen Leistungen zu bekräftigen.

Die signifikanten Unterschiede in der Selbstwirksamkeitserwartung in mathematischen Fähigkeiten bestätigt nicht nur Weinhardt (2017), sondern auch die Studie von Girnat et al. (2020). Zusätzlich zeigen sie noch, dass Mädchen der 11. Jahrgangsstufe in der Schweiz trotz des geringeren mathematischen Selbstkonzeptes vergleichbare Leistungen erbringen wie die der Jungen. Die Autor*innen sehen es deswegen als wichtig einen motivierenden Mathematikunterricht anzubieten, vor allem für Mädchen, um dadurch die Leistungsbereitschaft und das Interesse der Schülerinnen zu steigern. Genauso wird die Berücksichtigung von geschlechtsspezifischen Unterschieden hervorgehoben.

Die Studie von Guenaga (2022) untersuchte den Einfluss weiblicher Vorbilder in den Bereichen Wissenschaft, Technologie, Engineering oder Mathematik. Hierfür nahmen 303 Schüler*innen zwischen zehn und zwölf Jahren an einem Mentoring-

Programm teil. Nach der Teilnahme konnten eine Verbesserung gegenüber der Einstellung zur Technologie, die Anzahl der weiblichen Vorbilder in diesem Bereich, vor allem bei Mädchen festgestellt werden. Jedoch reichten die sechs Einheiten nicht aus, um einen signifikanten positiven Effekt in der mathematischen Selbstwirksamkeit festzustellen, ebenso wurden keine Veränderungen in den Geschlechterstereotypen bewirkt. Die Autor*innen begründen dies damit, dass Überzeugungen bezüglich der eigenen Fähigkeiten und Geschlechterstereotypen zu tief verankert sind, um in dieser kurzen Zeit Veränderungen bewirken zu können. Jedoch wird auch die geringfügige Verbesserung der Selbstwirksamkeit auf eine Art von Empowerment zurückgeführt, die im Mentoring-Programm wahrgenommen wurde.

2.4 Unterrichtskonzept

Das inklusive Unterrichtskonzept besteht aus den fünf einzelnen Stundenplanungen. Nach der Durchführung des Unterrichtskonzeptes sollten die Schüler*innen alle Lernziele erfüllt haben. Diese Lernziele werden durch Methoden, wie zum Beispiel Game Based Learning oder Pair-Programming umgesetzt. Zusätzlich zu den definierten Lernzielen aus dem schulautonomen Curriculum soll dieses Unterrichtskonzept auch die Selbstwirksamkeit im Programmieren, insbesondere von Schüler*innen, erhöhen. In diesem Abschnitt wird der theoretische Rahmen der Bausteine des Unterrichtskonzeptes erörtert.

2.4.1 Lernziele

Die Schüler*innen sollen verstehen, was ein Programm und ein Algorithmus, sowie ein Objekt ist. Des Weiteren sollen sie das Prinzip von Schleifen und Wenn-Funktionen verstehen und anwenden können. Dies alles soll schlussendlich in der bildungsorientierten blockbasierten Programmiersprache Scratch angewendet werden.

Das erste Lernziel besteht darin, dass die Schüler*innen verstehen, was ein Programm und ein Algorithmus sind. Nach Müller und Weichert (2017) ist ein Programm das Endresultat der Lösung eines Problems. Anfänglich existiert ein Problem mit einer Problemstellung. Mittels Abstraktion wird dadurch ein Algorithmus erstellt. Als letzter Schritt wird dieser Algorithmus in ein Programm transferiert. Durch die Realisierung wird dieses Programm anschließend an einem Rechner ausgeführt. Der Algorithmus ist das Herzstück. „Ein Algorithmus ist eine endliche Vorschrift zur

eindeutigen Überführung von Eingabegrößen zu Ausgabegrößen in einer endlichen Zahl von Schritten. (Müller & Weichert, 2017, S. 16)“ Die wesentlichen Eigenschaften eines Algorithmus sind nach Müller und Weichert (2017):

- Endlichkeit der Länge der Beschreibung
- Die Terminierung
- Eindeutige Reihenfolge der Operationen
- Eindeutige Wirkung der Anweisungen

Das Objekt wird nach Müller und Weichert (2017) als eine Datenstruktur definiert, bei dem es sich um eine konkrete Ausprägung einer Klasse handelt. Minnick et al. (2023) vergleichen zum besseren Verständnis das Objekt mit einem physischen, realen Ding wie der Gitarre. Eine Gitarre hat bestimmte Eigenschaften und verschiedene Nutzungsmöglichkeiten. Diese würden bei der objektorientierten Programmierung als Methoden betitelt werden.

Schleifen werden nach Minnick et al. (2023) verwendet, um die gleiche Anweisung mehrmals auszuführen, sprich zu wiederholen, und werden als Steueranweisung bezeichnet. Laut Müller und Weichert (2017) besteht die Ausführung der Schleife aus drei grundlegenden Schritten:

1. Auswertung der Bedingungen
2. Wenn die Bedingung erfüllt ist, wird die Anweisung ausgeführt und überprüft wieder Schritt eins
3. Bei Nicht-Erfüllung der Bedingung wird die Schleife terminiert.

Um Alternativen zu deklarieren, gibt es die „Bedingte Anweisung“ oder auch „Wenn-Funktion“ genannt. Müller und Weichert (2017) definieren den Aufbau bestehend aus dem Schlüsselwort „Wenn“ oder im Englischen „If“ gefolgt von einer Bedingung. Diese Bedingung hat nach Minnick et al. (2023) einen Booleschen Wert und kann somit „true/false“, wahr oder falsch sein. Je nachdem wird bei einem wahr, das darauffolgende ausgeführt oder bei einem falsch übersprungen.

All diese Grundlagen zum Programmieren werden im Unterrichtskonzept in der Programmiersprache Scratch schlussendlich angewendet. Diese Anwendung in Scratch bildet die letzten Lernziele, dass die Schüler*innen die Grundelemente der Programmierung anwenden und eine Geschichte in Scratch erstellen können.

Nach Scratch Foundation (2024) entwickelte die Lifelong Kindergarten Group am MIT Media Lab die Programmiersprache Scratch. Das Ziel ist „To spread creative, caring, collaborative, equitable approaches to coding and learning around the world (Scratch Foundation, 2024).“ Fildes (2007) beschreibt Scratch als ein Programm mit grafischer Benutzungsoberfläche, mit welcher die Anwender*innen Programme wie Bausteine zusammenstellen können. Diese Benutzungsoberfläche wird in Abbildung 4 dargestellt.

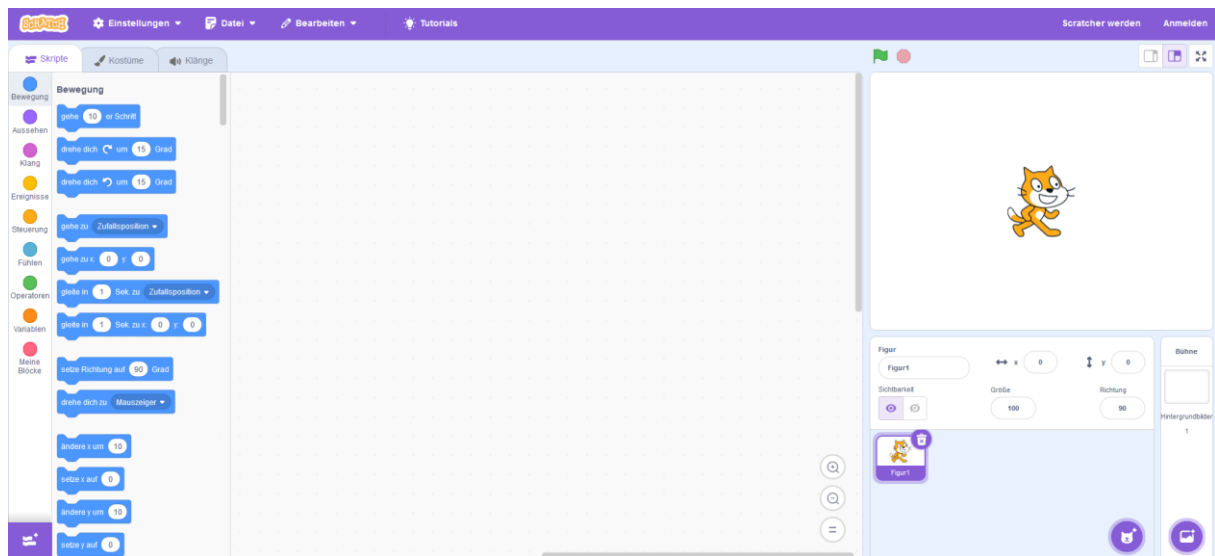


Abb. 4 Benutzungsoberfläche Scratch
(abgerufen von <https://scratch.mit.edu/projects/editor/?tutorial=getStarted>)

Durch diese sogenannten Bausteine wird ein Programm für eine Figur (Objekt), in diesem Fall die Katze, erstellt. Die Bausteine werden in Bewegung, Aussehen, Klang, Ereignisse, Steuerung, Fühlen, Operatoren, Variablen und Meine Blöcke untergliedert. Auf der Bedienoberfläche wird rechts oben mit Klicken der Startflagge das Programm ausgeführt.

2.4.2 Reflexion/Lerntagebuch

Nach Schön (1983/2016) wird immer und überall gelernt, doch um Lernen tiefergehend und nachhaltiger zu gestalten, ist die reflexive Praxis entscheidend.

Die Bedeutung Reflexion begründet Schön (1983/2016) folgendermaßen, dass technisches Wissen und Fertigkeiten nicht als einzige Grundlage professioneller Kompetenz reichen. Um komplexe Probleme zu lösen und die professionelle Kompetenz zu verbessern, ist die Reflexion, Improvisation und Anpassungsfähigkeit in der Praxis unumgänglich. Demnach ist das Modell des „Reflective Partitioner“ von

und nach Schön (1983/2016) notwendig um ein qualifizierter und selbstgesteuerter Professioneller, sowie eine selbstständig arbeitende Person zu werden.

Um diese Reflexion zu praktizieren, unterscheiden Esslinger-Hinz und Sliwka (2011) die professionelle Reflexion in drei Phasen: „Reflection-in-action“, „Reflection-on-action“ und „Reflection-for-action“. Bei der ersten Phase „Reflection-in-action“ wird direkt während des eigenen professionellen Handelns reflektiert. Im Gegensatz dazu wird bei der „Reflection-on-action“ erst direkt im Anschluss des Handels reflektiert. Die dritte und letzte Phase, damit die professionelle Reflexion stattfinden kann, ist laut Esslinger-Hinz und Sliwka (2011) die „Reflection-for-action“. Hier liegt der Fokus auf dem zukünftigen Handeln und wird in diesem Schritt im Voraus geplant. In der Unterrichtsplanung steht die Reflexion immer am Ende und ist direkt am Anschluss der Handlungen geplant. Deswegen entspricht die Reflexion einer „Reflection-on-Action“.

Umgesetzt wird dies im Unterrichtskonzept in der Form eines Lerntagebuches. Bräuer (2016) erläutert, dass das Tagebuch sich als Instrument eignet den Lernprozess zu dokumentieren, persönliche Einsichten festzuhalten sowie die Reflexionsfähigkeit zu fördern. Genauso kann es dazu beitragen, dass Studierende ihre Lernfortschritte besser verstehen und reflektieren können. Hervorgehoben wird, dass Herausforderungen auftreten können, wenn die Anleitung unzureichend ist.

Eine Möglichkeit Erfahrungen zu analysieren und reflektieren bietet der Reflexionskreislauf nach Gibbs (1988, zitiert nach (Esslinger-Hinz & Sliwka, 2011)). Dieser wird in der Abbildung 5 bildlich dargestellt.

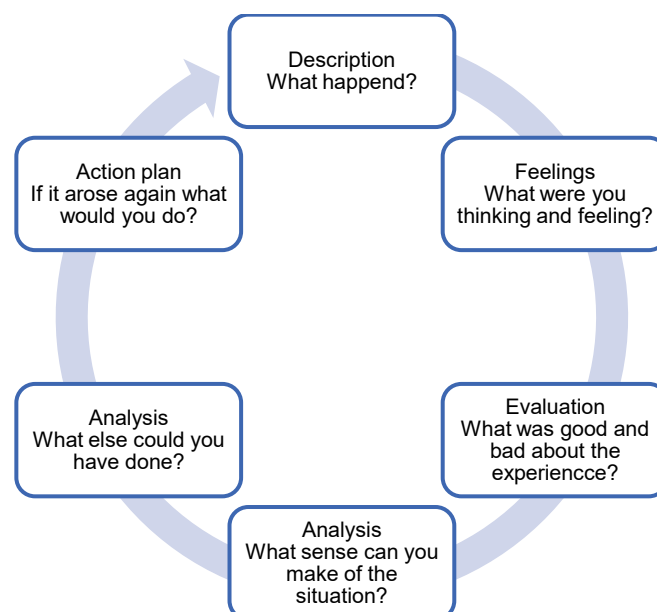


Abb. 5 Reflexionskreislauf nach Gibbs 1988 (Esslinger-Hinz & Sliwka, S. 33)

Nach Esslinger-Hinz und Sliwka (2011) sollen die Schüler*innen oder je nachdem die Reflektierenden als ersten Schritt jemanden oder sich selbst erklären, was genau in einer Situation passiert ist oder den zu reflektierenden Grund beschreiben. Im zweiten Schritt wird dann beschrieben was hierbei gefühlt und gedacht wurde. Anschließend wird evaluiert, was an der Situation gut und/oder auch schlecht war. Auf Grundlage dessen wird im nächsten Schritt Anstrengungen unternommen, den Sinn der Situation zu analysieren. Des Weiteren wird versucht im darauffolgenden Schritt alternative Handlungsmuster, Verhaltensweisen oder auch Reaktion zu finden. Im sechsten Schritt geht es um das zukünftige Verhalten. Wie sich zukünftig verhalten wird, wenn die Situation wieder eintreffen könnte, wird hier in diesem Schritt ein Aktionsplan erstellt. Da es sich um einen Kreislauf und nicht um einen Ablauf handelt, können all diese Schritte dementsprechend oft, wie vorkommend, wiederholt werden und mit dem Sinn sich somit kontinuierlich zu verbessern.

2.4.3 Game-Based Learning

Game-Based Learning kann verschiedene Formen annehmen, beginnend von einfachen Quizze bis hin zu interaktiven Abenteuern laut Meluso (2012). Es werden hierbei auf Spielen basierende Bildungstechnologien verwendet, die Spielelemente beinhalten, um Lerninhalte zu vermitteln und Lernprozesse zu unterstützen.

Chinchua (2022) erforschte die Steigerung der Programmier-Selbstwirksamkeit durch ein problemorientiertes Gamification-Digital-Learning-Ecosystem-Modell namens „The A3S3R DLE gamification learning model“. Das Modell bestand aus den sieben Schritten, beraten, zuweisen, analysieren, suchen, synthetisieren, zusammenfassen und berichten. Durch dieses Modell schafften sie es die Selbstwirksamkeit im Programmieren von thailändischen Studierenden zu verbessern. Dieses Erkenntnis bietet eine Grundlage, wie Game-Based Learning die Selbstwirksamkeit im Programmieren beeinflussen kann.

Genauso eine Grundlage, dass Game-Based Learning die Selbstwirksamkeit positiv beeinflussen kann liefert Lu und Lien (2020). Jedoch betonen sie auch, dass die Effekte unterschiedlich, je nach ihrer Wahrnehmung und Erfahrung in spielbasierten Lernumgebungen, sind. Die höchsten Effekte wurden bei Schüler*innen mit einer starken Wahrnehmung des Lernens und Spielens festgestellt. Im Gegensatz

dazu hatten die niedrigsten Effekte Schüler*innen mit einer spielorientierten Wahrnehmung erzielt.

Auch Meluso (2012) zeigte in seiner Studie, dass Games-Based Learning eine signifikante Auswirkung auf die Selbstwirksamkeit im MINT-Bereich hatte. Des Weiteren konnte erforscht werden, dass es keinen Unterschied macht, ob die Schüler*innen ein Einzel- oder Gruppenspiel absolvieren.

2.4.4 Pair-Programming

„Pair programming is a collaboration technique where two programmers work together to design, code and test the same programming task“ (Beck, 2000 zitiert nach Tan et al., 2023, S. 1061)

Nach Tan et al. (2023) arbeiten beim Pair-Programming zwei Programmierer*innen zusammen. Wei et al. (2021) unterschieden diese zwei Personen in Driver, beziehungsweise Fahrer*in, und Navigator*in. Der*die Fahrer*in führt die grundsätzliche Programmieraktion aus. Im Gegensatz dazu ist die Aufgabe der*des Navigator*in eine objektive Sichtweise zu behalten und das Bereitstellen der strategischen Denkweise für die Richtung der Arbeit.

Die zwei Autor*innenteams Tan et al. (2023) und Wei et al. (2021) stellten beide in ihrer Forschung eine signifikante Verbesserung der Selbstwirksamkeit beim Programmieren durch Pair-Programming fest.

Studierende der Universität China zeigten nach Tan et al. (2023) unterschiedliche Wirksamkeit in der positiven Veränderung der Selbstwirksamkeit je nach eingeteilten Gruppen. Es gab schnell und langsam arbeitende Studierende. Bei den schnell arbeitenden Studierenden gab es signifikante Verbesserungen der Selbstwirksamkeit im Programmieren, wenn sie mit dem Dialogmuster „Inquiry“ und „Guide“ arbeiteten. Bei dem Dialogmuster „Inquiry“ stellt der schnell arbeitende Studierende Fragen und Anleitungen, während der*die andere darauf reagiert und stellt somit einen interaktiven und gegenseitigen Wissenstransfer dar. Im Gegensatz dazu führten bei „Guide“ die schneller arbeitenden Studierende durch Fragen und Anleitungen die langsameren arbeitenden Studierenden. Im Gegensatz zu den langsam arbeitenden Studierenden gab es unabhängig vom Dialogmuster signifikante Veränderungen.

Die Ergebnisse der Studie von Palmer und Rao (2019) zeigen jedoch, dass die Ähnlichkeit der Selbstwirksamkeit zwischen Partner*innen einen signifikanten Einfluss auf den Lernerfolg der Studienteilnehmer*innen hat. Die Studie wurde an der Northern Arizona University mit den Studierenden der Lehrveranstaltung „Einführung in die Programmierung“ durchgeführt. Dort haben dann Paare mit ähnlicher Selbstwirksamkeit im Durchschnitt bessere Lernbewertungen erhalten als Paare mit unterschiedlicher Selbstwirksamkeit. Dies deutet darauf hin, dass die Selbstwirksamkeit der Teilnehmer*innen des Pair-Programming ein wichtiger Faktor für den Erfolg des Konzeptes ist.

Wei et al. (2021) stellten nicht nur eine signifikante Verbesserung in der Selbstwirksamkeit im Programmieren im Zusammenhang mit Pair-Programming, sondern auch Unterschiede zwischen Jungen und Mädchen fest. Insgesamt wurden 171 Grundschulschüler*innen untersucht. Die Autor*innen haben im Forschungsartikel darauf hingewiesen, dass die Jungen von dieser Pair Programming-Umgebung in der Selbstwirksamkeit mehr profitierten als die Mädchen. Sie betonen auch, dass es möglich ist, dass reine Mädchen-Teams, nicht zur Steigerung der Selbstwirksamkeit im Programmieren führen.

3. Planung der Aktionsforschung

In diesem Teil wird die Planung der Aktionsforschung, vor allem die theoriegeleitete Vorbereitung des Unterrichtskonzepts, erörtert. Das Konzept besteht aus insgesamt fünf Unterrichtseinheiten mit einer durchgehenden Reflexion in Form des Lerntagebuchs. Vor und nach der Durchführung des Unterrichtskonzeptes erhalten die Schüler*innen einen Fragebogen zur Selbstwirksamkeit beim Programmieren. Eine Unterrichtseinheit stellt eine Aktionsforschung mit zwei Zyklen dar. Diese wird als erstes geplant (welche in diesem Kapitel beschrieben wird) anschließend durchgeführt, die Ergebnisse gesammelt und analysiert (Kapitel Durchführung und Ergebnisse der Aktionsforschung), bevor sie dann überarbeitet und optimiert wird (Kapitel Reflexion und Analyse der Aktionsforschung). Dies wiederholt sich anschließend mit der zweiten Klasse. Diese letzte Überarbeitung spiegelt die Unterrichtseinheit wider, welche im schlussendlichen Konzept abgebildet wird.

3.1 Rahmenbedingungen

Die Aktionsforschung wird in insgesamt zwei Klassen der 7. Schulstufe in einem Gymnasium in Wien durchgeführt. In dieser Arbeit werden die Klassen mit Klasse A und Klasse B betitelt. Die Klasse A umfasst zwölf Schülerinnen und neun Schüler, deren Erziehungsberechtigte das Einverständnis zur Forschung erteilten. Ein*e Schüler*in der Klasse A wird nicht miteinbezogen. Die zweite Klasse (B) besteht aus 24 Schüler*innen, an der Forschung nahmen acht Schülerinnen und elf Schüler teil. Die Einverständniserklärung der Erziehungsberechtigten wurde mittels Elternbrief eingeholt.

Das Unterrichtsfach, indem die Forschung durchgeführt wird, heißt „Einführung in die Informatik“. Hier handelt es sich um ein schulautonomes Wahlpflichtfach. Die Schüler*innen des Gymnasiums können sich in der siebten Schulstufe für eine neue Sprache oder den Realzweig entscheiden. Im Realzweig bekommen sie anschließend jeweils eine Wochenstunde das Unterrichtsfach „Einführung in die Informatik“ in der siebten und achten Schulstufe.

Die Rahmenbedingungen gaben einen Zeitraum von fünfzig Minuten je Unterrichtseinheit vor. Insgesamt sind in der Jahresplanung fünf Unterrichtseinheiten für das Thema „Einführung in die Programmierung“ vorgesehen. Die Forschung fand in dem Zeitraum zwischen 22. Februar 2024 und 21. März 2024 statt.

Die Schüler*innen haben im schulischen Bereich kein Vorwissen bezüglich des Programmierens. Jedoch besteht die Möglichkeit, dass sie privat bereits Erfahrungen gesammelt haben.

3.2 Einverständniserklärung der Eltern

Zwei Wochen vor Beginn der Forschung erhielten die Eltern einen Elternbrief bezüglich meiner Forschung. In diesem Brief wurden vor allem kurz die Ziele der Masterarbeit beschrieben. Die durchgeführte Methode des Fragebogens sowie die Erhebung zu Beginn sowie am Ende der Forschung im Regelunterricht wurden erörtert. Zu Schluss wurde versichert, dass die Daten anonymisiert ausgewertet werden und ausschließlich wissenschaftlichen Zwecken dienen, sowie dass die Schulleitung bereits zugestimmt hat.

Die Eltern mussten einen Abschnitt der Einverständniserklärung mit ihrem Namen, den Namen des Kindes, Datum und Unterschrift ausfüllen. Ebenso war ein Häkchen mit der eindeutigen Zustimmung oder Absage der Teilnahme des Kindes zu der Forschung zu setzen. Diesen Abschnitt haben alle Kinder anschließend verpflichtend im Unterricht wieder abgeben müssen.

3.3 Fragebogen

Um den Ist-Stand bezüglich der Selbstwirksamkeit im Programmieren der Schüler*innen zu erheben wurde ein Fragebogen vor der Durchführung des Konzepts ausgeteilt. Der gleiche Fragebogen mit zwei zusätzlichen Freitextfragen wurde anschließend nach der Durchführung des Unterrichtskonzeptes verwendet.

Der Fragebogen basiert auf der „General Self-Efficacy Scale (GSE)“ von Schwarzer und Jerusalem (1995). Die „General Self-Efficacy Scale“ besteht aus insgesamt zehn Fragen und ist ein Selbstbeurteilungsverfahren. „Es misst die optimistische Kompetenzerwartung, das Vertrauen darauf, eine schwierige Lage zu meistern, wobei der Erfolg der eigenen Kompetenz zugeschrieben wird Schwarzer und Jerusalem (1999).“ Die Grundlage für diese Skala ist nach Schwarzer und Jerusalem (1999) das Selbstwirksamkeitskonzept.

Bei den Fragen gibt es vier verschiedene Antwortmöglichkeiten und für eine Antwortmöglichkeit verschiedene Punkte:

- Not at all true 1 Punkt
- Hardly true 2 Punkte
- Moderately true ... 3 Punkte
- Exactly true 4 Punkte

Insgesamt können bei zehn Fragen zwischen zehn und vierzig Punkten erreicht werden. Je höher das Ergebnis ausfällt, umso höher ist die Selbstwirksamkeit. Die Reliabilität des ursprünglichen Fragebogens liegt laut Schwarzer und Jerusalem (1995) bei 76% bis 90%. Dies deutet darauf hin, dass die Skala als zuverlässig anzusehen ist, um die wahrgenommene Selbstwirksamkeit zu messen.

Für diese Arbeit wurden die Fragen auf Deutsch übersetzt und anschließend immer auf das Programmieren beziehungsweise die Vorübungen für das Programmieren bezogen. Es wurde beispielsweise aus „I can always manage to solve difficult problems if I try hard enough (Schwarzer & Jerusalem, 1995)“ zu „Ich schaffe es, schwierige Probleme beim Programmieren zu lösen, wenn ich mich ausreichend anstrenge.“

Als Testdurchgang füllten vier 13-jährige Schüler*innen aus Oberösterreich den Fragebogen aus und gaben zusätzlich noch ein kurzes schriftliches Feedback. Den zweiten Satz fand eine der Schüler*innen etwas unverständlich, weswegen er nochmals leicht abgeändert wurde. Ansonsten verstanden sie alle Fragen leicht beziehungsweise gut. Insgesamt gab es somit, vom ursprünglichen Fragebogen ausgegangen, zehn neue, auf das Programmieren bezogene Fragen.

1. Ich schaffe es, schwierige Probleme beim Programmieren zu lösen, wenn ich mich ausreichend anstrenge.
2. Wenn ich beim Programmieren auf Hindernisse stoße, kann ich Mittel und Wege finden, um mein Ziel zu erreichen.
3. Es fällt mir leicht, an meinen Zielen beim Programmieren festzuhalten und diese zu erreichen.
4. Ich bin zuversichtlich, dass ich mit unerwarteten Ereignissen beim Programmieren effizient umgehen kann.
5. Dank meines Einfallsreichtums weiß ich, dass ich mit unvorhergesehenen Situationen beim Programmieren umgehen kann.

6. Ich kann die meisten Probleme beim Programmieren lösen, wenn ich die nötige Anstrengung aufbringe.
7. Ich kann ruhig bleiben, wenn es beim Programmieren schwierig wird, weil ich mich auf meine Bewältigungsfähigkeiten verlassen kann.
8. Wenn ich mit einem Problem beim Programmieren konfrontiert werde, kann ich meist mehrere Lösungen finden.
9. Wenn ein Programm nicht funktioniert, kann ich normalerweise eine Lösung finden.
10. Ich kann mit allem umgehen, was beim Programmieren auf mich zukommt.

Das Selbstbeurteilungsverfahren mit dem Punktesystem blieb gleich. Zur besseren Lesbarkeit wurde nicht der Gesamtscore von zehn bis vierzig Punkten errechnet, sondern ein Durchschnittswert, welcher zwischen einem und vier Punkten liegen kann. Der in dieser Masterarbeit verwendete gesamte Fragebogen ist im Anhang zu finden.

Den ausgedruckten Fragebogen bekamen die Schüler*innen in der ersten Unterrichtseinheit am Anfang, sowie in der letzten Unterrichtseinheit am Ende erneut zum Ausfüllen. Die Schüler*innen gaben diesen anschließend wieder der Lehrperson ab. Der Fragebogen in der letzten Unterrichtseinheit beinhaltete zwei weitere Freitextfragen

- Inwiefern spielt für dich die Lehrperson eine Rolle bei der Begleitung beim Programmieren-Lernen?
- Hast du während der letzten Unterrichtsstunden eine Veränderung deines Vertrauens in deine Fähigkeiten beim Programmieren bemerkt? Wenn ja, beschreibe welche Faktoren dazu beigetragen haben?

Die erste Frage hat den Zweck zu erforschen, ob die Schüler*innen die Lehrperson als Vorbild beim Programmieren-Lernen wahrnehmen. Im Gegensatz zur zweiten Frage, welche auf die eigenen Einschätzungen der Schüler*innen in der Selbstwirksamkeit im Programmieren abzielt. Des Weiteren soll die Frage auch jene Faktoren erforschen, welche ihrer Meinung eine positive Wirkung gehabt haben, um die Selbstwirksamkeit im Programmieren zu stärken.

3.4 Erste Unterrichtseinheit: Programm, Objekt, Schleifen

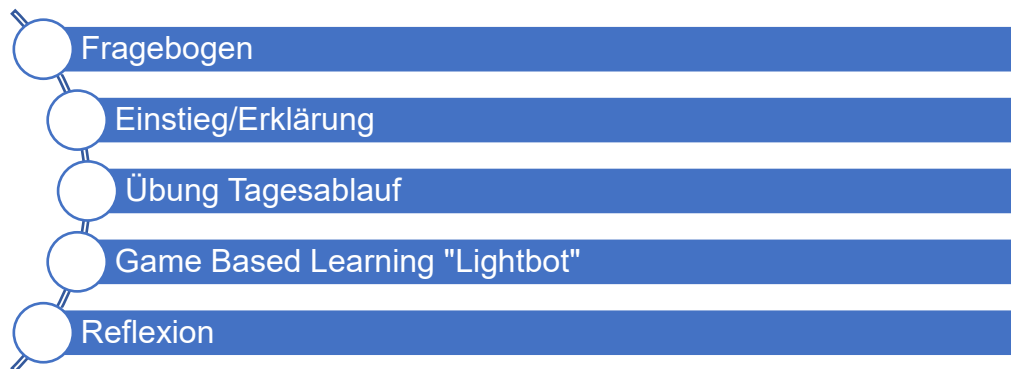


Abb. 6 Übersicht erste Unterrichtseinheit

Nach dem Ausfüllen des Fragebogens beginnt die erste Unterrichtseinheit des Konzeptes. Das Ausfüllen des Fragebogens beansprucht circa zehn Minuten, weswegen für die restliche Einheit noch vierzig Minuten übrigbleiben.

Anfänglich wird nur ein Strichmensch auf das Whiteboard gezeichnet und die Schüler*innen dürfen einen Namen für diesen Strichmensch bestimmen. Anschließend werden gemeinsam mit den Schüler*innen typische Tagesabläufe erarbeitet, ohne dabei ein spezifisches Thema zu nennen. Es muss nur darauf geachtet werden, dass eine Wiederholung (zum Beispiel Unterricht – Essen – Unterricht – Essen) vorhanden ist.

Im Anschluss werden die Schüler*innen gefragt, wie dies mit dem Programmieren zusammenhängen könnte. Mit dem Hintergrund, dass die Schüler*innen erkennen, dass ein Programm schlussendlich abstrakt dargestellt, ein Ablauf ist mit einer Terminierung, eindeutigem Beginn und eindeutiger Reihenfolge der Operatoren.

Da später mit einer objektorientierten Programmiersprache gearbeitet wird, wird den Schüler*innen die Frage gestellt: „Für wen gilt dieser Tagesablauf“. Die Antwort wird der Strichmensch sein und somit wird das Thema Objekt eingeführt.

Für die Einführung der Schleifen, sollen die Schüler*innen jene Stelle finden, die sich immer wiederholt. Anschließend wird diese Stelle mit Unterricht/Essen mit zweimaliger Wiederholung (x2) zusammengefasst und damit die Schleife eingeführt. Bevor dies an das Whiteboard geschrieben wird, sollen die Schüler*innen selbst erkennen, wie oft die Schleife wiederholt werden soll.

Jetzt kommen die Schüler*innen in die erste Arbeitsphase und haben die Aufgabe einen eigenen Tagesablauf in einem Word Dokument zu schreiben mit einem Objekt, als auch einer Wiederholung drinnen. Diese Wiederholung sollte einmal mit und einmal ohne Schleife dargestellt werden. Die Abgabe erfolgt über die Lernplattform Eduvidual.

Nach diesem Input kommt es zum ersten Game-Based Learning Abschnitt. Die Schüler*innen sollen das Spiel Lightbot auf der Webseite lightbot.lu spielen. Das originale Konzept von Lightbot stammt von LightBot Inc. Die Webseite wurde von Larent Haan entwickelt, das Interface von Zenobia Homan gestaltet und "surt" als auch "hektikmusic" waren auch an der Webseite beteiligt (Haan Laurent). Es handelt sich um einen Roboter, der mittels Anweisungen zu dem blauen Feld zu bewegen ist, welches der Roboter erleuchten muss. Zur Auswahl stehen die Anweisungen „Walk forward, Turn 90° to the right, Turn 90° to the left, Jump, Light, Repeat“. Durch Ziehen der Anweisungsblöcke können die Schüler*innen ihr Programm erstellen.

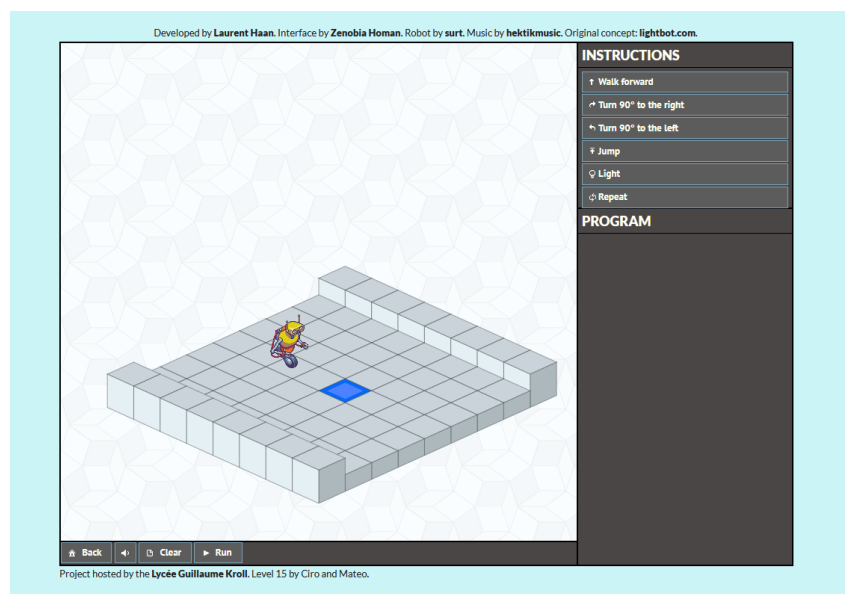


Abb. 7 Lightbot Level 0 (abgerufen von <https://lightbot.lu/>)

Je nachdem, wie viele Anweisungen benötigt werden, bekommt man keine beziehungsweise eine bronzene, silberne oder goldene Medaille. Da die Schüler*innen Schleifen üben sollen, bekommen sie als Vorgabe, bei den Levels mindestens eine silberne Medaille zu erreichen. Da mit einer Schleife weniger Anweisungen benötigt werden und somit ein besseres Ergebnis und höherwertige Medaille erzielt werden kann.

In den letzten zehn Minuten haben die Schüler*innen Zeit ihren ersten Eintrag im Lerntagebuch auf der Lernplattform Eduvidual zu erstellen. Um die Reflexion, die die Selbstwirksamkeit beim Programmieren beeinflussen kann, zu vereinfachen, erhielten die Schüler*innen folgende zwei Leitfragen:

- Beschreibe einen Zeitpunkt im Unterricht, bei dem du dich besonders herausgefordert gefühlt hast. Wie hast du diese Herausforderung gemeistert und welche Strategien hast du angewendet, um erfolgreich zu sein?
- Reflektiere über einen Moment während des Programmierens oder Vorübungen des Programmierens, indem du ein Erfolgserlebnis hattest. Was hat dazu beigetragen, dass du dich kompetent und selbstwirksam gefühlt hast, und wie kannst du dieses Gefühl in zukünftigen Programmieraufgaben nutzen?

3.5 Zweite Unterrichtseinheit: Wenn-Funktionen

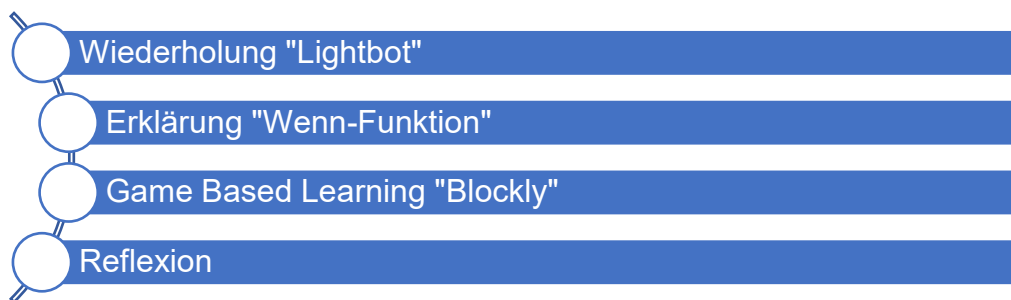


Abb. 8 Übersicht zweite Unterrichtseinheit

Nach einer kurzen mündlichen Wiederholung des Gelernten (Objekt und Schleifen) der letzten Einheit, bekommen die Schüler*innen zehn Minuten Zeit das Spiel Lightbot noch einmal zu spielen. Dies bietet den Schüler*innen die Möglichkeit sich individuell zu verbessern und auch zu steigern.

Anschließend wird die Wenn-Funktion erarbeitet. Es wird ein neuer Tagesablauf auf die Tafel geschrieben, der den Tagesablauf eines Wochenendes darstellt. Nun sollen die Schüler*innen den Unterschied finden (Schule findet statt oder nicht).

Um die Wenn-Funktionen anzuwenden, bekommen die Schüler*innen die Aufgabe Blockly zu spielen. Blockly Games ist eine Webseite von Code with Google. Blockly Games besteht aus mehreren Spielen, geeignet für Kinder ohne Programmiererfahrungen (Code with Google). Ein Spiel von Blockly Games bildet das Labyrinth. „Maze is an introduction to loops and conditionals. It starts simply, but every

level is more challenging than the last (Code with Google).“ Somit eignet das Spiel sehr gut, um die Schleifen und Wenn-Funktionen gleichzeitig zu üben.

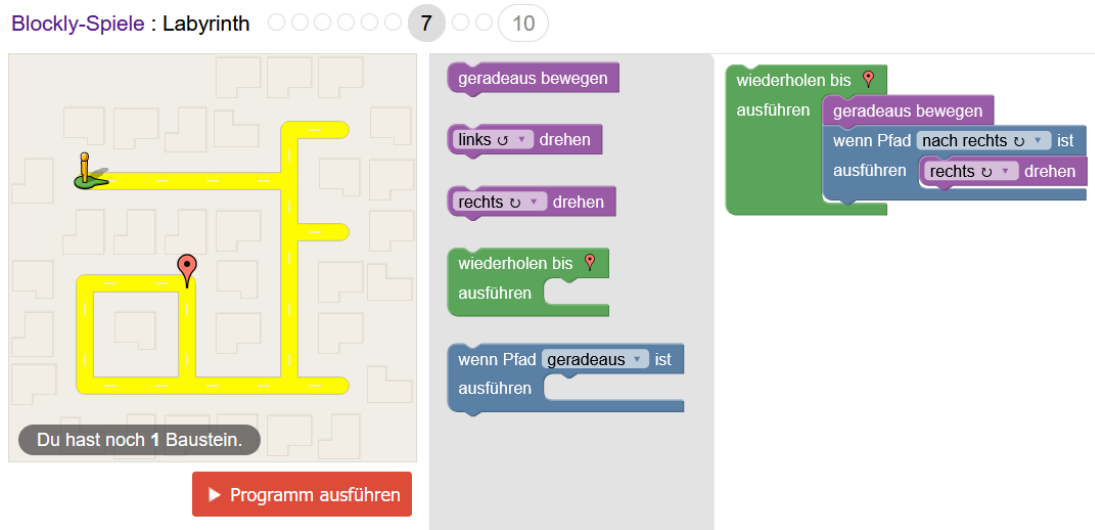


Abb. 9 Spiel Blockly Level 7 (abgerufen von <https://blockly.games/maze?lang=de&level=7&&skin=0>)

Wie bei Lightbot gibt es Blöcke zum Zusammenfügen, um ein Programm zu erstellen. Das Programm kann bei dem Button „Programm ausführen“ gestartet werden und der kleine Mensch beginnt die Anweisungen zu befolgen. Als Erstes müssen nur Bewegungen programmiert werden und mit steigenden Levels kommen zuerst Schleifen beziehungsweise schlussendlich auch Wenn-Funktion hinzu. Der Vorteil an diesem Spiel ist, dass es eine Begrenzung hat, wie viele Blöcke verwendet werden dürfen. Somit werden die Schüler*innen gezwungen, Schleifen und Wenn-Funktionen zu verwenden.

Die Reflexion im Lerntagebuch wurde im Vergleich zur ersten Einheit abgeändert nach dem Reflexionskreislauf nach Gibbs (1988) nach Esslinger-Hinz & Sliwka (2011). Die Anweisung auf der Lernplattform Eduvidual war nun:

Überschrift des Lerntagebuch-Eintrages: **Wenn Funktionen**

Erster Absatz Herausforderung

1. Beschreibe einen Zeitpunkt im Unterricht, indem du dich herausgefordert gefühlt hast.
2. Was hast du dabei gedacht oder gefühlt?
3. Was war gut und was war schlecht bezüglich der Erfahrung?
4. Welchen Sinn kannst du aus der Situation ziehen?
5. Was hättest du anders machen können?
6. Wie kannst du diese Erfahrung in zukünftigen Programmieraufgaben nutzen?

Zweiter Absatz Erfolgserlebnis

- Reflektiere über einen Moment während des Programmierens oder Vorübungen des Programmierens, in dem du ein Erfolgserlebnis hattest. Was hat dazu beigetragen, dass du dich kompetent und selbstwirksam gefühlt hast, und wie kannst du dieses Gefühl in zukünftigen Programmieraufgaben nutzen?

Abb. 10 Lerntagebuch-Angabe auf der Lernplattform Eduvidual

Für den Lerntagebucheintrag sind in der Planung wieder zehn Minuten Zeit vorgesehen.

3.6 Dritte Unterrichtseinheit: Pair-Programming in Scratch

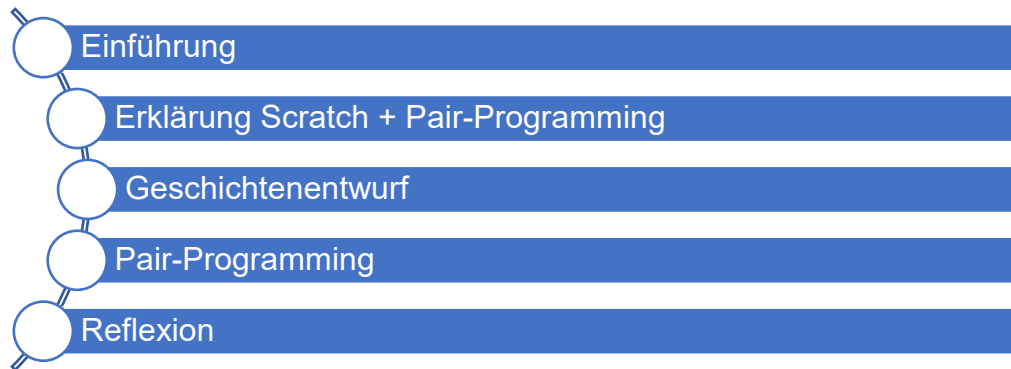


Abb. 11 Übersicht dritte Unterrichtseinheit

Während die ersten zwei Einheiten darauf abzielen, die Grundlagen der Programmierung kennenzulernen, ändert sich jetzt der Fokus auf die Anwendung. Das Ziel ist es nun, dass die Schüler*innen eine interaktive Geschichte mit Scratch programmieren zu können.

Als Einstieg wird den Schüler*innen eine solche interaktive, programmierte Geschichte auf Scratch gezeigt. Um den Schüler*innen Vorbilder zu liefern beziehungsweise die stellvertretende Erfahrung zu aktivieren, beinhaltet die Geschichte Ada Lovelace. Um den Inhalt der Geschichte zu erstellen, wurde die Künstliche Intelligenz ChatGPT verwendet. Es stellte sich folgender Dialog als Ergebnis heraus.

Ada Lovelace: Hey Computer, wie läuft's heute?

Computer: Hi Ada! Mir geht's gut, danke. Was kann ich für dich tun?

Ada Lovelace: Weißt du noch, wie wir zusammen die ersten Programme geschrieben haben? Das war echt aufregend, oder?

Computer: Oh ja, das war eine spannende Zeit! Du hast damals schon verstanden, dass ich nicht nur Zahlen berechnen kann, sondern auch komplexe Aufgaben bewältigen kann.

Ada Lovelace: Genau! Es war faszinierend zu entdecken, was wir zusammen erreichen können, indem wir die richtigen Anweisungen geben.

Computer: Deine Ideen haben die Grundlage für das moderne Programmieren gelegt. Du hast gezeigt, dass mit Kreativität und Durchhaltevermögen alles möglich ist.

Ada Lovelace: Danke, Computer. Es war toll, diesen Weg mit dir zu gehen. Ich hoffe wirklich, dass viele Mädchen und Frauen, inspiriert durch unsere Geschichte, ihre eigenen Fähigkeiten im Programmieren entdecken und ausbauen können.

Computer: Ich bin sicher, dass sie es können. Unsere Geschichte zeigt, dass jeder, egal welches Geschlecht oder welchen Hintergrund er hat, die Welt der Technologie mitgestalten kann.

Dieser Dialog zwischen Ada Lovelace und dem Computer wurde anschließend, wie auf Abbildung 12 gezeigt, auf Scratch umgesetzt.

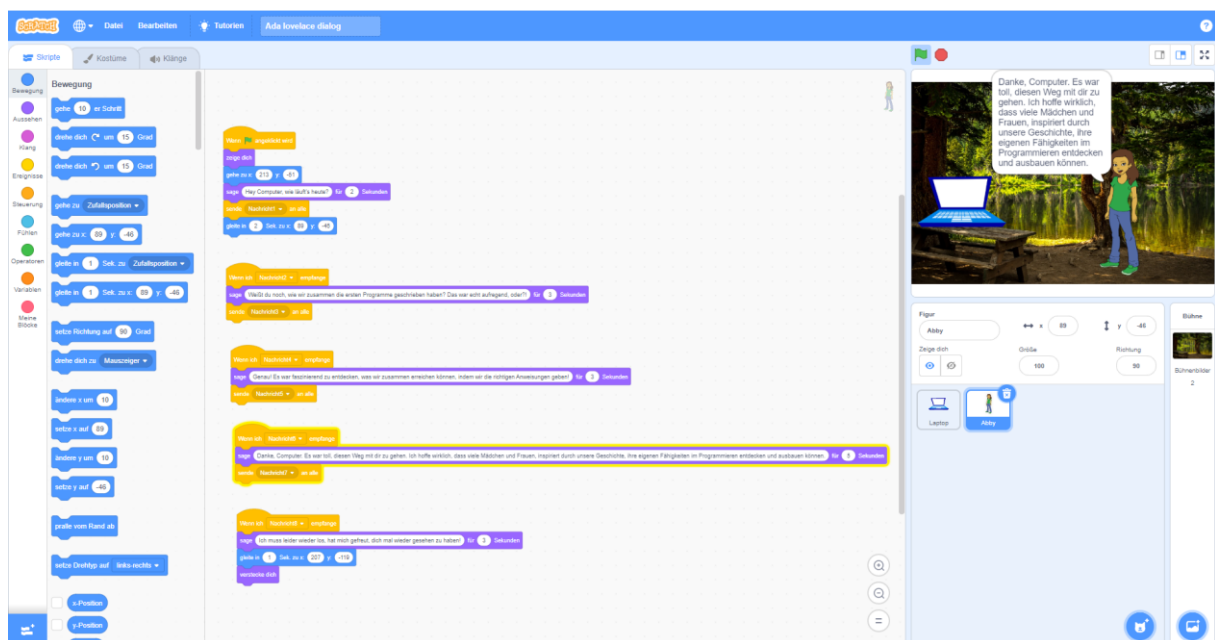


Abb. 12 Beispiel Geschichte von Ada Lovelace und Computer auf Scratch

Nach dem Einstieg bekommen die Schüler*innen allgemeine Erklärung wie Scratch funktioniert. Vor Allem wird auf die Objekte, Bühnenbilder, das Koordinatensystem sowie die verschiedenen Befehle eingegangen. Erklärt wird dies alles anhand eines Beispiels mit der Katze und einem Apfel, welches mit den Schüler*innen gemeinsam von Grund auf selbst live programmiert wurde.

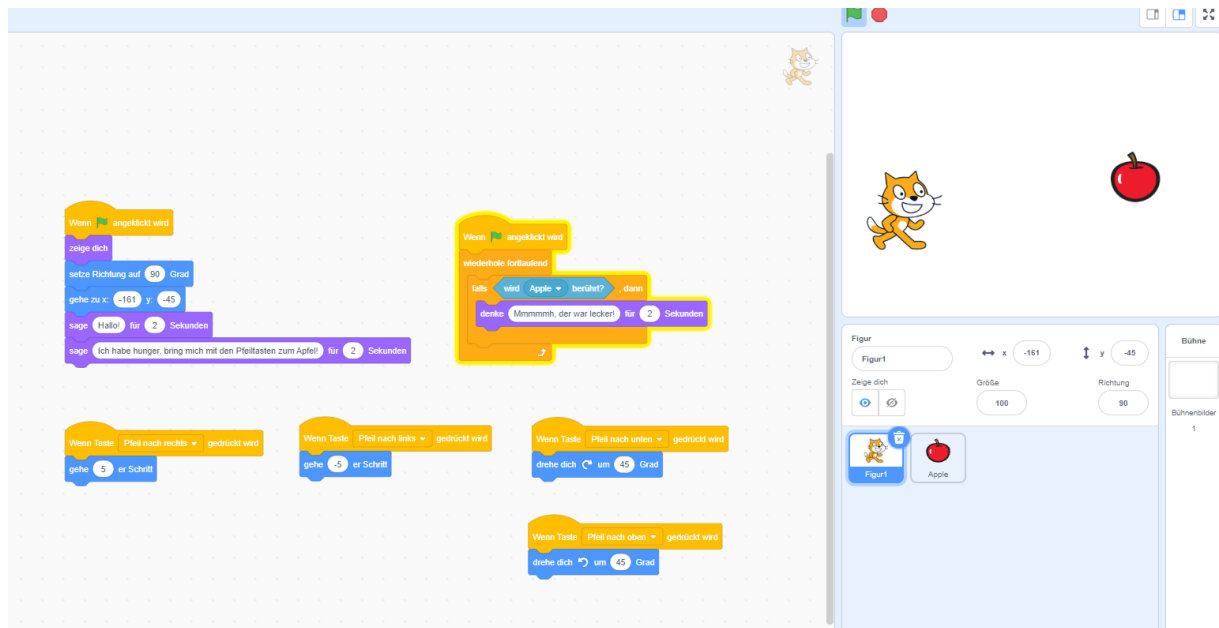


Abb. 13 Beispiel zur Erklärung

Anschließend bekamen die Schüler*innen die Erklärung anhand einer Präsentation, was Pair-Programming bedeutet und wie es funktioniert, sowie die genaue Aufgabenstellung:

Partnerarbeit (Pair-Programming): Erstellung einer eigenen Story in Scratch!

Aufgabe
Einstellungen
Erweiterte Bewertung
Mehr ▾

Erstelle in einer Partnerarbeit (Pair-Programming) eine eigene Story in Scratch!

Erster Schritt: Überlegt euch eine geeignete Story und beschreibt diese schriftlich in ein paar Sätzen.

Mindestanforderungen: Auswahl des Bühnenbildes, zwei Figuren, Bewegung mit Richtung und Geschwindigkeit, zwei Dialoge. Orientiert euch an den Beispielen von Storytelling (siehe Link).

Bitte gebt diese Beschreibung als Textdatei ab!

Zweiter Schritt: Erstellt die Story in Scratch im Pair-Programming. Jeder übernimmt eine Rolle – entweder bist du der Driver oder der Navigator. Das ist etwa so wie beim Rallye fahren, wo jemand fährt und der andere die Karte liest und ansagt, welche Kurve als nächstes kommt. Der Driver entwickelt das Programm und bedient den Computer. Der Navigator überlegt was als nächstes zu tun ist, denkt über Problemstellungen nach, kontrolliert das entstehende Programm und macht Vorschläge. Die beiden Rollen werden regelmässig getauscht.

Erstellt ein Scratch-Projekt, speichert es und gebt es als Datei ab!

Hilfestellungen.pdf
5. März 2024, 10:12

Abb. 14 Aufgabenstellung Pair-Programming

Im Anschluss sehen die Schüler*innen ein Beispiel einer Schülerin des Vorjahres. Dies soll eine stellvertretende Erfahrung erzeugen, da die Schülerin für die Schüler*innen greifbarer ist. Dies führt, wie im theoretischen Rahmen erwähnt, nach Kriegler-Kastelic (2018) zu höheren Effekten.

Nach dieser Erklärungsphase bekommen die Schüler*innen fünf Minuten Zeit, ihre Idee für die interaktive Geschichte in einer Word-Vorlage zu dokumentieren.

Anschließend beginnt die Pair-Programmierung der Geschichte auf Scratch. Alle fünf Minuten ist ein Wechsel zwischen Driver und Navigator*in. Die Arbeitsphase dauert insgesamt zwanzig Minuten.

Zum Abschluss bekommen die Schüler*innen wieder zehn Minuten Zeit, um den Lerntagebucheintrag mit folgenden zwei Leitangaben zu erstellen:

Reflektiere über einen Moment während des Programmierens oder Vorübungen des Programmierens in dieser Unterrichtseinheit, indem du ein Erfolgserlebnis hattest, und beschreibe diesen genau. Wann hattest du ein Erfolgserlebnis? Was hat dazu beigetragen, dass du dich kompetent und selbstwirksam gefühlt hast, und wie kannst du dieses Gefühl in zukünftigen Programmieraufgaben nutzen?

Falls du kein Erfolgserlebnis hattest, beschreibe einen Zeitpunkt im Unterricht, indem du dich herausgefordert gefühlt hast. Was war gut und was war schlecht bezüglich der Erfahrung? Welchen Sinn kannst du aus der Situation ziehen? Was hättest du anders machen können? Wie kannst du diese Erfahrung in zukünftigen Programmieraufgaben nutzen?

3.7 Vierte Unterrichtseinheit: Pair-Programming in Scratch



Abb. 15 Übersicht vierte Unterrichtseinheit

Die vierte Unterrichtseinheit verläuft ähnlich der dritten Unterrichtseinheit. Die in der letzten Einheit anwesenden Schüler*innen dürfen nach einer kurzen mündlichen Wiederholung des Unterrichtsstoffes der letzten Einheit mit dem Pair-Programming beginnen. Im Gegensatz dazu begann für die anderen Schüler*innen die geplante dritte Unterrichtseinheit.

In dieser Unterrichtseinheit soll die Lehrperson den Fokus auf die Schüler*innen und individuellen Feedbacks setzen.

Auch der Fokus des Lerntagebuchs wurde für die Reflexion dieser Einheit geändert. Die Schüler*innen bekommen zusätzlich den Auftrag über die Rolle und das Geschlecht der Lehrperson als Vorbild zu reflektieren:

1. Denke darüber nach, ob deine Lehrperson ein Vorbild für dich ist, wenn es darum geht, Programmieren zu lernen, und wie das deine Zuversicht beeinflusst, dass du es schaffen kannst. Macht es einen Unterschied für dich, ob deine Lehrperson weiblich oder männlich ist? Schreibe deine Gedanken dazu auf.

2. Beschreibe einen Moment im Unterricht, in dem du durch deinen Glauben an dich selbst etwas erreicht hast. Was lief gut und was lief weniger gut in dieser Situation? Welche Bedeutung kannst du aus dieser Erfahrung ziehen? Gibt es Dinge, die du in Zukunft anders machen würdest? Wie kannst du diese Erfahrung in zukünftigen Programmieraufgaben nutzen?

3.8 Fünfte Unterrichtseinheit: Präsentation und Feedback



Abb. 16 Übersicht fünfte Unterrichtseinheit

Als Einstieg wurden die Merkmale des guten Feedbacks wiederholt (die Schüler*innen kennen diese aus den anderen Unterrichtsfächern). Merkmale werden auf der Tafel von der Lehrperson mitnotiert.

Anschließend beginnt die Feedbackrunde. Die Schüler*innen öffnen ihre fertige Geschichte am PC. Die Schüler*innen teilen sich aus dem Pair-Programming Team in eine*n Präsentator*in und Zuseher*in auf. In Folge bleibt die Präsentator*in am Tisch und präsentiert die Geschichte, währenddessen geht der*die Zuseher*in einen Tisch weiter wechselt. Dort schaut er*sie sich eine andere Geschichte an und gibt dazu Feedback. Nach fünf Minuten wechselt der*die Zuseher*in einen Tisch weiter, bis sie mit allen Geschichten durch ist. Darauffolgend wechseln Zuseher*in und Präsentator*in die Rollen.

Schlussendlich füllen die Schüler*innen ihren letzten Eintrag im Lerntagebuch aus. Die Reflexion zielt wieder auf die Steigerung der Selbstwirksamkeit durch die Programmierung und anschließenden Präsentation der Scratch-Geschichte mit folgender Angabe ab:

Welcher Moment während der Präsentation deiner Scratch-Geschichte hat dir gezeigt, dass du in der Lage bist, Probleme zu lösen und deine Ideen erfolgreich umzusetzen? Wie hat dich dieser Moment in deinem Glauben an deine Fähigkeit gestärkt, als Programmierer*in Herausforderungen zu bewältigen?

In den letzten zwanzig Minuten bekommen die Schüler*innen Zeit, um den abschließenden Fragebogen (Fragebogen 2) der Forschung auszufüllen.

4. Durchführung und Ergebnisse der Aktionsforschung

Die Durchführung der Aktionsforschung fand von Februar 2024 bis März 2024 statt. In diesem Teil werden die durchgeführten Einheiten sowie die Beobachtungen erörtert. Zwischen Durchgang eins und Durchgang zwei wurden die Unterrichtseinheit beziehungsweise deren Ergebnisse reflektiert und verbessert. Diese können im nächsten Kapitel „Reflexion und Analyse der Aktionsforschung“ gefunden werden.

4.1 Erste Unterrichtseinheit – Schleifen

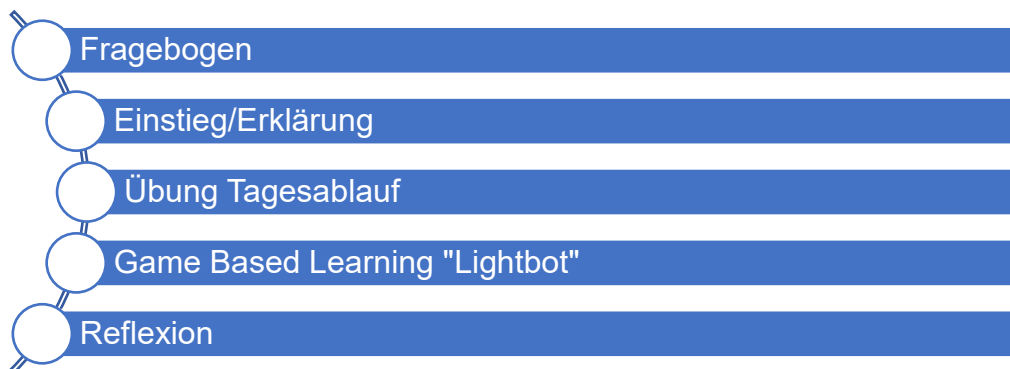


Abb. 17 Übersicht erste Unterrichtseinheit

4.1.1 Durchgang 1

Die erste Unterrichtseinheit wurde am 22. Februar 2024 in der Klasse A durchgeführt. Es waren alle Schüler*innen anwesend, davon zwölf Mädchen und neun Jungen.

Nach dem Einstieg bekamen die Schüler*innen als erste Aufgabe den Fragebogen zur Selbstwirksamkeit beim Programmieren auszufüllen. Ein Schüler hob gleich am Anfang die Hand und fragte nach, wie es zu verstehen ist, denn sie können noch nicht programmieren. Daraufhin kam die Erklärung für die ganze Klasse, dass gemeint wäre, wie sie sich einschätzen später beim Programmieren agieren zu können. Sobald sie fertig waren, bekamen sie die Anweisung mit ihren Sesseln nach vorne zu kommen, um mit dem eigentlichen Teil des Unterrichts zu starten.

Zur Mitbestimmung und Aktivierung durften die Schüler*innen sich einen Namen für das Strichmenschlein auf der Tafel überlegen. Die Schüler*innen wurden aufgefordert den Tagesablauf einer Schulwoche, in ihrem Fall für „Kevin“, zu erarbeiten. Mehr als zwei Drittel der Schüler*innen haben mitgearbeitet und verschiedene Punkte aufgezählt, welche alle auf der Tafel notiert wurden. Die einzigen Punkte „Schule – Essen – Schule – Essen“ wurden vorgegeben. Anschließend kam

die Frage „Wie könnte man das kompakter schreiben“ und es kam sofort die Antwort „als Wiederholung“ von mehreren Schüler*innen. Daraufhin wurde einmal Schule-Essen weggestrichen und stattdessen einmal Schule-Essen mit „2x“ notiert. Hier fand anschließend der Übergang zum Programmieren mit der Erklärung statt, dass das Objekt in diesem Fall Kevin ist und das Programm für Kevin geschrieben worden ist. Ebenfalls wurden die Schüler*innen darauf hingewiesen, dass das erste Konstrukt der Schleife erarbeitet wurde. Anschließend bekamen die Schüler*innen die Aufgabe einen Tagesablauf für sich mit mindestens einer zusätzlichen Schleife in Word zu schreiben und die Schüler*innen begaben sich deswegen wieder zu ihren Computern.

Die Schüler*innen arbeiteten anschließend leise in Einzelarbeit an ihren PCs und führten die Aufgabe ohne weitere Fragen oder sichtbaren Schwierigkeiten aus.

Folgend wurden sie aufgefordert das Spiel „Lightbot“ zu spielen. Man konnte gut beobachten, wie die Schüler*innen mit Begeisterung versuchten den Roboter mit den passenden Anweisungen zum Ziel zu bringen. Weiteres bekamen sie auch die Anweisung mindestens eine Silbermedaille pro Level zu erreichen. Ein Schüler fragte, ob er jetzt eine Schleife in einer Schleife verwenden darf. Dies wurde gelobt und bestätigt, dass dies genau der richtige Ansatz ist und er so weitermachen sollte. Anschließend kamen von dem Schüler keine weiteren Fragen mehr und er löste alle zukünftigen Aufgaben allein und der Beobachtung nach mit Zuversicht.

Es waren allerdings nur mehr ungefähr sieben Minuten für Lightbot übrig, da die anschließende Reflexion in der Form eines Lerntagebuchs noch am Programm stand. Die Enttäuschung des Abbruches des Spiels brachten die Schüler*innen mit einem kollektiven „maaaaa“ zum Ausdruck.

Die Reflexion führten die Schüler*innen in den letzten sieben Minuten wieder in Einzelarbeit auf ihren Computern durch. Hier konnte man allerdings beobachten, wie vereinzelte Schüler*innen sich Inspiration bei ihren Nachbar*innen holten. Diese wurden daraufhin aufgefordert, diese allein zu erledigen. Des Weiteren konnten drei Schüler*innen die Reflexion aufgrund technischer Probleme nicht abgeben.

Die Reflexionsfragen waren:

- Beschreibe einen Zeitpunkt im Unterricht, bei dem du dich besonders herausgefordert gefühlt hast. Wie hast du diese Herausforderung gemeistert und welche Strategien hast du angewendet, um erfolgreich zu sein?

- Reflektiere über einen Moment während des Programmierens oder Vorübungen des Programmierens, indem du ein Erfolgserlebnis hattest. Was hat dazu beigetragen, dass du dich kompetent und selbstwirksam gefühlt hast, und wie kannst du dieses Gefühl in zukünftigen Programmieraufgaben nutzen?

Nachdem die Schüler*innen die Reflexion beendeten, durften sie die letzten Minuten nochmal ihr Glück beim Spiel „Lightbot“ versuchen, bevor die Stunde zu Ende war.

4.1.2 Durchgang 2

In der Klasse B wurde die erste Unterrichtseinheit am 27. Februar 2024 durchgeführt. Aufgrund der Abwesenheit eines Schülers waren insgesamt nur acht Mädchen und zehn Jungen anwesend.

Aufgrund einer mündlichen Prüfung und sonstiger Angelegenheiten konnte der Fragebogen erst nach zwanzig Minuten ausgefüllt werden. Sobald sie fertig waren, wurden sie angewiesen sich in die Mitte zu setzen und dort zu warten. Währenddessen wurde das Strichmenschlein auf dem Whiteboard gezeichnet und die Überschrift „Tagesablauf“ aufgeschrieben.

Als alle Schüler*innen sich in der Mitte versammelt haben, war die erste Aufgabe einen Namen für den Strichmenschen auf der Tafel zu finden. Das Ergebnis war der Name „Pablo“. Beim Erstellen der Tagesablauf-Punkte waren durchgängig ca. 5-10 Hände oben, um einen Punkt nennen zu können. Bei der Frage zum Übergang von Tagesablauf zur Programmierung heben wieder fünf Schüler*innen die Hände. Die Erarbeitung der Schleife und des Objekts geschah auch wieder durch Mitarbeit der Schüler*innen. Zwei Schüler erkannten auch, dass man eigentlich alle Punkte in eine Schleife mit einer Wiederholung von 5x geben könnte.

Die Schüler*innen bekamen anschließend, wie bei der Klasse A, die Aufgabe ihren eigenen Tagesablauf zu erstellen. Der Unterschied war dieses Mal die Verwendung der Vorlage. Die Vorlage beinhaltete die Aufgabenstellung (mindestens zehn Tagesablauf-Punkte und eine neue Schleife) sowie ein Feld zum Ausfüllen des Tagesablaufes ohne und mit Schleife. Zum Schluss mussten die Schüler*innen am Aufgabenblatt beantworten, wer/was das Objekt ist beziehungsweise für wen der Tagesablauf ist und welche Verwendung Schleifen haben. Die Aufgabe wurde dann zum Schluss auf Eduvidual hochgeladen. Circa die Hälfte der Schüler*innen konnte

schlussendlich noch mit Lightbot beginnen, die anderen waren bis zum Ende der Einheit mit dem Tagesablauf beschäftigt.

4.2 Zweite Unterrichtseinheit – Wenn-Funktionen

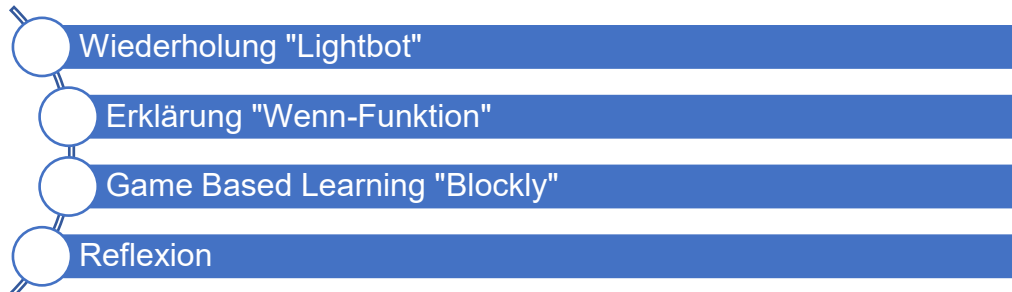


Abb. 18 Übersicht zweite Unterrichtseinheit

4.2.1 Durchgang 1

Die zweite Unterrichtseinheit fand am 29. Februar 2024 statt und es fehlte ein Schüler, weswegen insgesamt zwölf Mädchen und acht Jungen anwesend waren.

Der Start der Unterrichtseinheit war eine mündliche Wiederholung des Stoffes der letzten Woche und anschließend das Weiterspielen von Lightbot. Die Schüler*innen bekamen zehn Minuten Zeit, um weitere Level zu meistern. Den Beschwerden nach, als sie zu einem Ende kommen mussten, war die Motivation für dieses Spiel hoch.

Die Erarbeitung der Wenn-Funktionen gemeinsam an der Tafel erfolgte flott, da die Schüler*innen die Fragen zur Erarbeitung schnell beantworten konnten. Ein Lernender zog die Assoziation zu Scratch (die Erfahrung wurde privat gesammelt, in der Schule war dies noch nicht Unterrichtsstoff), dass es hier auch Wenn-Funktionen gäbe. Es konnte erklärt werden, dass es sich hier um eine allgemeine Funktion handelt, die in mehreren Programmiersprachen existiert.

Als die Schüler*innen anschließend die Wenn-Funktionen ausprobieren konnten, gab es keine Fragen und das Level sechs (ab diesen musste Wenn-Funktionen angewendet werden) wurde von jeder*jedem Schüler*in gemeistert.

Bei der Reflexion beantworteten alle Schüler*innen nur die erste Aufgabenstellung mit den Fragen zur Herausforderung im Unterricht. Kein*e Schüler*in beantwortete die zweite Aufgabenstellung.

4.2.2 Durchgang 2

Die zweite Unterrichtseinheit der Klasse B fand am 5. März 2024 statt und es waren sieben Mädchen und zehn Jungen anwesend.

Die Wiederholung der letzten Einheit funktionierte schnell, die Schüler*innen konnten alle Punkte der letzten Einheit aufzählen. Anschließend spielten sie zum ersten Mal das Spiel Lightbot, da es sich in der letzten Stunde nicht ausgegangen war. Als Erstes haben manche Schüler*innen das Prinzip Schleifen zu verwenden, nicht verstanden, nach kurzer Erinnerung wandten es allerdings alle an.

Die Erarbeitung der Wenn-Funktionen verlief, wie auch bei der Klasse A, einwandfrei. Die Intervention, eigene Wenn-Funktionen-Beispiele zu überlegen, führte zu einem Beispiel im realen Leben. Das Beispiel war aus dem Bereich kochen: „Wenn die Kartoffel zehn Minuten gekocht hat, dann aus dem Topf geben (Schüler)“.

Anschließend spielten die Schüler*innen Blockly. Die Intervention der Peer-Sitzungen musste nicht erwähnt werden, da sich die Schüler*innen im Vorfeld schon begannen sich gegenseitig zu unterstützen und sich über die verschiedenen Level auszutauschen. Dies fand immer statt, wenn sie darauf warteten, dass ihr Programm ausgeführt wird und zu Ende kommt.

4.3 Dritte Unterrichtseinheit – Pair Programming

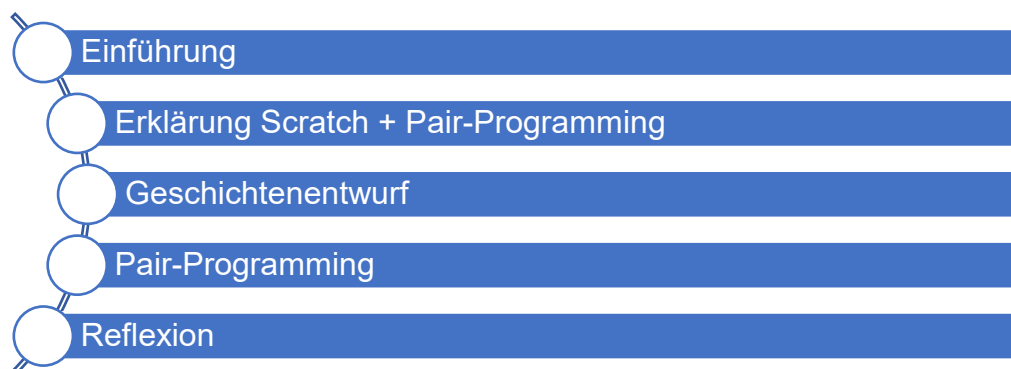


Abb. 19 Übersicht dritte Unterrichtseinheit

4.3.1 Durchgang 1

Die Einheit fand am 7. März 2024 statt und es waren 18 Schüler*innen anwesend. Insgesamt besuchten somit neun Mädchen und acht Jungen der Klasse A den Unterricht.

Zum Einstieg wurde die interaktive Geschichte von Ada Lovelace präsentiert bei dem die Schüler*innen in der Mitte gesessen sind und aufmerksam der Geschichte folgten. Anschließend wurde das heutige Lernziel angekündigt, die Programmierung einer interaktiven Geschichte. Dies führte gleich zum zweiten Punkt, und zwar zur Erklärung von Scratch. An der Tafel wurden Objekte, Bühnenbilder, das Koordinatensystem und die verschiedenen Befehle erläutert. Anschließend wurden all diese Punkte an einem praktischen Beispiel präsentiert, wo die Katze meint, sie hat Hunger und sie soll mit den Pfeiltasten zum Apfel gebracht werden. Sobald sie den Apfel erreicht hat, fehlt am Apfel ein kleines Stück.

Im Anschluss wurden die Mindestanforderungen der Geschichte besprochen. Anschließend wurde noch ein Beispiel einer Schülerin des letzten Jahres hergezeigt. Anhand der PowerPoint wurden das Konzept und die Regeln des Pair-Programming erklärt. Keine*r der Schüler*innen kannte davor das Konzept des Pair-Programming. Vier Schüler*innen wollten die Teams wechseln, jedoch nach kurzer Erklärung akzeptierten sie die Teameinteilung und begannen zu arbeiten.

Dann begann die Teameinteilung nach dem Score, den sie beim Fragebogen am Anfang des Unterrichtskonzepts ausgefüllt haben. Schüler*innen, die in dieser Einheit fehlten, wurden ausgelassen und werden in der nächsten Einheit jeweils zwei Paare bilden.

Nach einer fünf-minütigen Erarbeitungsphase der Geschichte ging auch das Pair-Programming los. Es wurde nach fünf Minuten gewechselt und insgesamt waren für den Programmierteil noch 15 Minuten über, sprich insgesamt zwei Wechsel zwischen Driver und Navigator. Zum Abschluss gab es wieder die Reflexion bestehend aus der Reflexion eines Erfolgserlebnisses oder einer Herausforderung in dieser Einheit.

4.3.2 Durchgang 2

Der zweite Durchgang fand in der Klasse B am 12. März 2024 statt. Es waren acht Schülerinnen und neun Schüler anwesend.

Der Einstieg war wieder die Präsentation der vorprogrammierten Geschichte mit Ada Lovelace. Gefolgt von der Erklärung von Scratch mit dem Unterschied, bevor erklärt wurde, wurden die Schüler*innen gefragt, wie verschiedene Anweisungen etc. funktionieren. Es kam immer eine Antwort von den Schüler*innen. Insgesamt halfen

fünf verschiedene Schüler*innen, indem sie Erklärungen für die Fragestellungen als Erstes lieferten, bevor die Lehrperson nochmal zusammenfasste.

Anschließend folgte wie im ersten Durchgang die Erklärung für Pair-Programming, sowie die Präsentation des Beispiels einer Schülerin des Vorjahres.

Nach der Einteilung dem Ranking folgend, mittels des in der ersten Einheit durchgeführten Fragebogens, begannen die Schüler*innen zu zweit mit ihrem kurzen Geschichtsentwurf. Kurz beschwerten sich Schüler*innen durch Raunen, dass die Teams vorgeben sind. Jedoch nach Erklärung, worauf die Einteilung beruht, akzeptierten die Schüler*innen diese Entscheidung.

Das Pair-Programming wurde wieder im fünf-Minuten-Modus durchgeführt, sprich alle fünf Minuten wurde gewechselt. Insgesamt sind sich somit vier Durchgänge ausgegangen, bevor die Schüler*innen zur wöchentlichen Reflexion im Lerntagebuch wechselten.

4.4 Vierte Unterrichtseinheit – Pair Programming



Abb. 20 Übersicht vierte Unterrichtseinheit

4.4.1 Durchgang 1

Am 14. März 2024 fand die vierte Unterrichtseinheit mit dem Schwerpunkt auf Pair-Programming in der Klasse A statt. Es fehlten zwei Schüler*innen. Insgesamt waren deswegen 19 Schüler*innen anwesend.

Nachdem kurz die letzte Stunde wiederholt wurde, gab es eine neue Einteilung des Pair-Programming. Der Grund war, dass die zwei fehlenden Schüler*innen aus verschiedenen Teams waren und die Teamkolleg*innen somit neu zugeteilt wurden. Nach kurzer Klärung wer Driver und wer Navigator*in ist, starteten die Schüler*innen wieder mit dem Programmieren in Scratch. Nach jeweils fünf Minuten wurde immer gewechselt.

Anfänglich bekamen die Schüler*innen, welche letzte Stunde gefehlt haben, die Einführung in Scratch und Pair-Programming. Anschließend ging die Lehrperson zu

den verschiedenen Gruppen und half bei Verständnisfragen. In einem Durchgang mussten die Schüler*innen ihr Spiel kurz präsentieren und bekamen daraufhin gezieltes Feedback, was sie gut gemacht haben, für kreative Ideen und erfolgreiche Umsetzung dieser. Dreimal wollten die Schüler*innen verschiedene Fragen zum Programmieren beantwortet haben, wurden ermutigt es selbst zu probieren und schafften dies schlussendlich auch.

In den letzten zehn Minuten bekamen die Schüler*innen wieder Zeit für ihr Lerntagebuch. Dieses Mal lag der Schwerpunkt bezüglich des Einflusses der Lehrperson als Vorbildfunktion, aber auch um ein Erfolgserlebnis in dieser Einheit.

4.4.2 Durchgang 2

Der zweite Durchgang fand dieses Mal am Montag, dem 18. März 2024 in der Klasse B statt. In dieser zusätzlichen Supplierstunde außerhalb der Einheit am Dienstag waren zwölf Schüler*innen anwesend, da sechs Personen fielen.

Nach einer kurzen Wiederholung zur Benutzung von Scratch und der Aufgabenstellung bekamen die Schüler*innen diese Einheit Zeit, um an ihrem Scratch Projekt weiterzuarbeiten.

Da Schüler*innen krank waren, mussten Teams neu gemischt werden. Insgesamt kam es somit zu fünf neuen Paarungen. Bei diesen Paarungen konnte beobachtet werden, dass das Pair-Programming Schwierigkeiten aufzeigte und somit wurden zwei dieser neuen Paarungen insgesamt viermal ermahnt sich an die Rollen beziehungsweise vereinbarten Regeln zu halten.

Der Modus wurde in dieser Einheit in zehnminütigen Intervallen gewechselt, anstatt fünf Minuten. Es wurden jedoch keine sichtbaren Veränderungen beobachtet.

In den letzten zehn Minuten wurden die Schüler*innen wieder aufgefordert sich dem Lerntagebucheintrag dieser Einheit zu widmen, welcher die Schüler*innen auch nachgekommen sind.

4.5 Fünfte Unterrichtseinheit – Präsentation und Feedback



Abb. 21 Übersicht fünfte Unterrichtseinheit

4.5.1 Durchgang 1

Aufgrund der zusätzlichen Unterrichtseinheit fand dieses Mal der erste Durchgang am 19. März 2024 in der Klasse B statt. Es waren 17 Schüler*innen anwesend.

Gleich am Anfang gab es eine Änderung, die von der Planung abwich. Anstatt den Schüler*innen zu erklären, was gutes Feedback ausmacht, wurde dieses gemeinsam erarbeitet. Die erste Antwort war, es soll darauf eingehen, ob die Anforderungen, wie zum Beispiel ein Dialog, erfüllt worden sind. Anschließend wurden auf der Tafel deswegen die Stichpunkte „auf Mindestanforderungen eingehen“ notiert. Allerdings wurde der Punkt anfänglich von mir falsch aufgefasst und somit kam eine neue Sichtweise heraus, und zwar dass das Feedback im Dialog stattfinden sollte. Als die Schüler*innen sich danach nicht mehr meldeten, wurde gefragt, ob man den anderen beleidigen darf, welches von den Schüler*innen verneint worden ist und „respektvoll“ wurde somit notiert. Anschließend warfen die Schüler*innen noch ein, es soll „spezifisch“ und „konstruktiv“ sein, womit fast alle Punkte aus der Planung abgedeckt wurden. Eine weitere kurzfristige Änderung war, dass die Schüler*innen den Auftrag bekamen, zu versuchen mindestens eine Stärke konkret im Feedback zu erwähnen.

Im Anschluss begann die Feedback-Runde. Nachdem ein Feedback beobachtet, wurde zwischen zwei Schülerinnen, welches aus „gut“ und nicht mehr bestand, wurde die Schülerin darauf hingewiesen, zu sagen, was genau gut ist. Dies benötigte drei Anläufe, bevor ein spezifisches Feedback zustande kam. Es gab auch Verwirrung, wie der Wechsel stattfinden sollte. Es wurde anschließend von der Lehrperson versucht verschiedenen Personen verschiedene Geschichten zuzuweisen.

Die Hälfte der Klasse musste anschließend aufgrund eines Lehrausfluges den Unterricht verlassen und werden den abschließenden Fragebogen in einem anderen Unterrichtsfach am selben Tag noch ausfüllen.

Für den Rest ging es wieder an das Lerntagebuch, bevor sie anschließend die abschließenden Fragebogen ausfüllten. Bei dem Lerntagebuch war dieses Mal folgende Fragen zu beantworten: „Welcher Moment während der Präsentation deiner Scratch-Geschichte hat dir gezeigt, dass du in der Lage bist, Probleme zu lösen und deine Ideen erfolgreich umzusetzen? Wie hat es sich angefühlt? Wie hat dich dieser Moment in deinem Glauben an deine Fähigkeit gestärkt, als Programmierer*in Herausforderungen zu bewältigen?“

4.5.2 Durchgang 2

Bei der letzten Einheit des Unterrichtskonzeptes waren zwanzig Schüler*innen anwesend. Die Einheit fand am 21. März 2024 in der Klasse A statt.

Als Erstes wurden wieder gemeinsam mit den Schüler*innen Kriterien eines guten Feedbacks erarbeitet. Die Schüler*innen nannten als ersten Punkt das Sandwich-Feedback und auf Nachfrage meinerseits wurde darauf mit folgenden zwei Kriterien „auf Stärken eingehen“ sowie „Verbesserungsvorschläge bringen“ eingegangen, sowie diese auch auf der Tafel notiert. Anschließend kamen noch „konstruktiv“ und „ehrlich“. Um der Entwicklung entgegenzuwirken, dass nur „gut“ aus Feedback kommt, wurde mit den Schüler*innen erarbeitet, dass es spezifisch sein soll. Die zweite Intervention wurde anschließend umgehend umgesetzt, dass sie sich die Geschichte mit der Ada Lovelace nochmal ansahen und anschließend gefragt wurden, wie hier ihr Feedback aussehen würde. Als Erstes kamen ausschließlich negative Argumente, die daraufhin gemeinsam in konstruktive Verbesserungsvorschläge umgewandelt wurden. Gegen Ende kamen dann auch positive Argumente und Stärken zum Vorschein.

Bei der Feedbackrunde wurden die Schüler*innen von einem Zweier-Team immer einem anderen Team zugewiesen. Eine Person von Team A erklärte die Geschichte einer Person von Team B und zeitgleich auch genau einmal andersrum. Anschließend zeigte die Person, die vorher Feedback gegeben hatte von Team B, der präsentierenden Person von Team A die Geschichte und dementsprechend weiter. Nach einem solchen Feedback Zyklus bekamen die Zweier-Teams noch einmal ein neues Team zugeteilt, mit dem identen Ablauf. Die Schüler*innen verstanden dieses

Mal das System und setzten es dementsprechend problemlos um. Auch das Feedback, bei dem ich zuhören konnte, war ausführlich und den Kriterien entsprechend.

Zu Stundenende bekamen die Schüler*innen noch Zeit, für ihren letzten Lerntagebucheintrag mit der gleichen Frage wie bei Durchgang eins, sowie um den abschließenden Fragebogen auszufüllen.

5. Reflexion und Analyse der Aktionsforschung

In diesem Abschnitt werden die gesammelten Daten des vorherigen Kapitels analysiert und nach den Unterrichtseinheiten und Durchgängen reflektiert. Durch Identifizierung der Stärken und Schwächen, sowie Herausforderungen oder überraschenden Ergebnissen werden die Interventionen bewertet und gegebenenfalls überarbeitet oder auch weitere Adaptionen für zukünftige Einheiten getroffen.

5.1 Erste Unterrichtseinheit – Schleifen

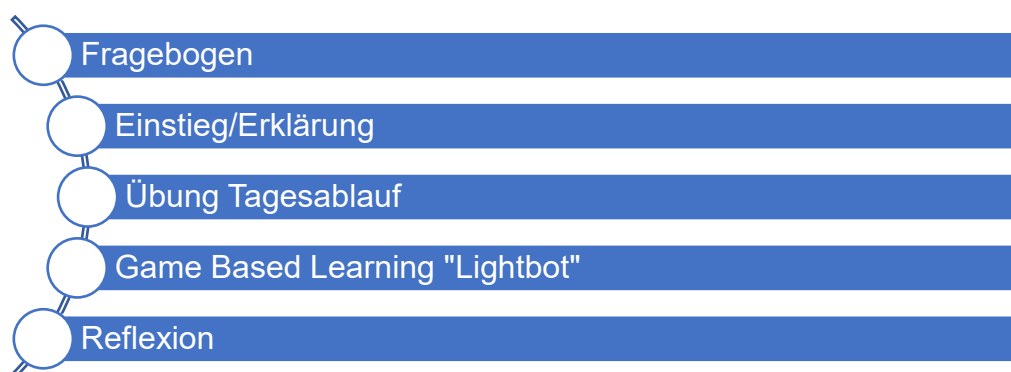


Abb. 22 Übersicht erste Unterrichtseinheit

5.1.1 Durchgang 1

Die Einführung und Erarbeitung des Stoffes haben gut funktioniert. Durch die maßgebliche Mitbestimmung der Schüler*innen konnte erkannt werden, dass sich ihre Motivation steigerte. Dies zeigt sich, indem sie mehrere meldeten und noch mehr hinzufügen wollten und gebremst werden mussten, damit der Tagesablauf nicht zu lang wird.

Mit der ersten Einzelaufgabe wirkten die Schüler*innen fast unterfordert. Deswegen wird dieser Punkt angepasst werden und der nächsten Klasse ein Arbeitsblatt ausgeteilt. Dieses beinhaltet die gleiche Aufgabe, jedoch mit dem Zusatz das Objekt genau zu definieren und der Frage, was genau eine Schleife ist. Da dies einfach für die Schüler*innen war, konnten allerdings alle Erfolge feiern, welches sich hoffentlich positiv auf die Selbstwirksamkeit auswirken wird.

Das Spiel „Lightbot“ kam bei den Schüler*innen gut an. Es konnte beobachtet werden, wie sie mit Motivation versuchten die Aufgaben zu lösen und mehrere Wege ausprobierten, um das Problem zu meistern. Manche setzten sich ebenso das Ziel eine Goldmedaille zu erhalten. Vor Allem hier wurde in dieser Einheit von der Lehrperson auf unterstützendes Feedback geachtet. Der Beobachtung nach lernten

die Schüler*innen als Problemlösestrategien, das Problem in kleine Probleme zu unterteilen. Das konnte erkannt werden, da sie öfters erst mal nur einen Teil programmierten und ausprobierten, um zu sehen, ob dieser funktioniert und anschließend den Rest programmierten. Genauso wie die Strategie einfach auszuprobieren, verändern und anschließend zu verbessern.

Bei der ersten Frage im Lerntagebuch besteht Verbesserungsbedarf, da sechs Schüler*innen nur kurz mit „gar nicht“ und „niemals“ antworteten. Dies lässt darauf schließen, dass bei denjenigen Schüler*innen keine tiefgreifende Reflexion stattgefunden hat beziehungsweise damit die Zielintuition dieser Aufgabe nicht erreicht worden ist. In der nächsten Klasse wird diese Frage mit dem Reflexionskreislauf nach Gibbs (1988) im Lerntagebuch modifiziert werden.

Der Fokus der zweiten Frage war die Steigerung der Selbstwirksamkeit. Hier waren diesbezüglich positive Reflexionen dabei, welche Faktoren zur Beeinflussung der Selbstwirksamkeit widerspiegeln. Die Schüler*innen gingen vor allem auf das Spiel „Lightbot“ ein. Eine Schülerin schrieb:

*„Bei Lightbot, wenn ich in der Zukunft ein Problem bei etwas
ähnlichem habe könnte ich mich daran erinnern und so das Gefühl
bekommen das ich es schon einmal geschafft habe und nicht aufgeben
soll.“ (Schülerin)*

Bei dieser Schülerin wird deswegen davon ausgegangen, das Ziel erreicht zu haben. Sie gibt an ein Problem gehabt zu haben und darauf jetzt eine gute Grundlage baut, weitere Probleme erfolgreich lösen zu können, was genau auf die Definition der Selbstwirksamkeit nach Bandura (1997) zutrifft. Weitere fünf Schüler*innen beschrieben das Gefühl von Motivation und 13 von 19 Schüler*innen, dass sie ein Erfolgserlebnis hatten.

5.1.2 Durchgang 2

Leider ging durch die Prüfung etc. mehr Zeit verloren wie geplant, was dazu führte, dass sich nicht die ganze Planung ausging.

Der Einstieg verläuft für diese Klasse gut, da normalerweise immer nur ein*e Schüler*in, öfters noch ein/zwei andere sich melden. Dies könnte folgendermaßen interpretiert werden, dass die gemeinsame Erarbeitung des Stoffes die Schüler*innen als motivierend empfunden haben. Positiv war auch die Erkenntnis zweier

Schüler*innen, dass der gesamte Tagesablauf schlussendlich in eine Schleife gepackt werden könnte.

Eine gute Intervention war das vorgefertigte Arbeitsblatt zum Tagesablauf mit Schleifen. Alle Schüler*innen konnten die Schleifen anwenden, bei zwei Schüler*innen fehlte allerdings die Definition, wie oft die Schleife wiederholt werden sollte. Als Verbesserung könnte die Wichtigkeit des genauen Arbeitens beim Programmieren/Programmier-Vorübungen erwähnt werden. Auch die Fragen auf dem Arbeitsblatt wurden fehlerfrei ausgefüllt und damit das fachliche Ziel erreicht.

Äußerst schade war es, dass durch die Umstände verkürzte Unterrichtsstunde die Schüler*innen zwar in den letzten fünf Minuten Zeit für die Reflexion gehabt hätten, jedoch noch tief im Abschluss beziehungsweise der Abgabe des Tagesablaufes verstrickt waren. Nur ein Schüler hat es schlussendlich geschafft die Reflexion auszufüllen.

Da in der nächsten Einheit sowieso eine Wiederholung beziehungsweise der Einstieg mit Lightbot geplant war, werden die Schüler*innen Zeit für die Reflexion und Lightbot bekommen.

5.2 Zweite Unterrichtseinheit – Wenn-Funktionen

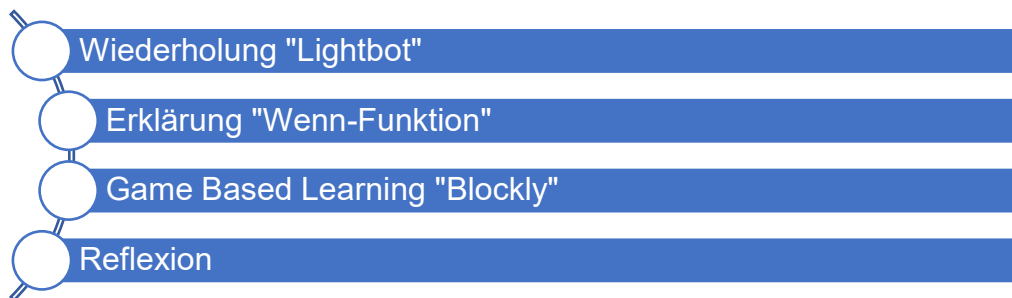


Abb. 23 Übersicht zweite Unterrichtseinheit

5.2.1 Durchgang 1

Nach wiederholtem Einsatz des Spiels Lightbot und Betrachtung der Reflexionen, wurde davon ausgegangen, dass die Schüler*innen sogenannte aktive Meistererfahrungen sammeln konnten. Eine Schülerin schrieb in der Reflexion über Lightbot bei der Frage „Was war gut und was war schlecht bezüglich der Erfahrung“ und „Wie kannst du diese Erfahrung in zukünftigen Programmieraufgaben nützen?“ folgende Antwort:

„3. Es ist gut auch Hindernisse beim Programmieren zu haben, da kann man lernen geduldig und Ausdauer zu haben... 6. Ich kann mich an diesen Moment erinnern wenn ich Schwierigkeiten habe, um geduldig zu bleiben.“ (Schülerin)

Die Schülerin beschreibt jene Kriterien, dass es sich nach Bandura (1997) um keinen einfachen Erfolg handelt, sondern mit einer Herausforderung verknüpft ist. Ähnlich hat eine Schülerin geschrieben „... 4. Es hat geholfen das ich es aus allen Blickwinkeln betrachtet habe ... (Schülerin)“. Ein weiterer Schüler hat die Herausforderung ebenfalls als „gute Erfahrung“ deklariert. Genauso eine weitere Schülerin schrieb, dass sie dadurch nächstes Mal weiß, wie es funktioniert und dies auch im späteren Leben hilfreich sein kann. Durch den spielerischen Effekt wirkt es, erkennbar an den Beschwerden zum Beenden des Spieles, zusätzlich auch noch motivierend.

Das Erarbeiten der Wenn-Funktion hat inhaltlich gesehen gut funktioniert. Es wird aufgrund der fehlenden Reflexionen davon ausgegangen, dass bezüglich der Selbstwirksamkeit keine Effekte erzielt worden sind. Weshalb dies für die nächste Stunde angepasst wird. Die Schüler*innen sollen aufgefordert werden, selbst mögliche „Wenn-Funktionen-Situationen“ zu überlegen und sich mündlich zu melden.

Beim Blockly-Spiel haben sich die Schüler*innen den Reflexionen nach, vor allem beim letzten Level, Level 10, herausgefordert gefühlt. In der Reflexion ist dies von zwanzig Schüler*innen zehn Mal genannt worden. Von den zehn Schüler*innen konnten sieben Schüler*innen daraus positive Rückschlüsse ziehen. Am meisten wurde genannt, dass sie nicht aufgeben sollten und es dann geschafft werden könnte. Dies lässt auch wieder auf eine aktiv stattgefundene Meistererfahrung schließen. Eine mögliche Intervention aufgrund der aus der Literatur bekannten Einflussfaktoren wird beim nächsten Mal eine Peer-Feedback-Sitzung eingebaut. Bei den Peer-Feedback-Sitzungen werden die Schüler*innen aktiv aufgefordert, einander zu unterstützen und ihren Mitschüler*innen konstruktive Rückmeldungen zu geben.

Die Intervention der Reflexion auf den Reflexionskreislauf nach Gibbs (1988) nach Esslinger-Hinz und Sliwka (2011) anzupassen war eine gute Maßnahme, da die Schüler*innen mehr in der Tiefe reflektierten. Dies lässt sich daraus ableiten, weil keine Antworten mehr wie „nichts“ kamen.

Ein Problem bei diesem Konzept ist die durch die Rahmenbedingungen vorgegebene Zeit. Für die Reflexion bekommen die Schüler*innen in diesem Fall zehn Minuten Zeit. Eine Intervention wäre die einzelnen 50-Minuten-Einheiten auf Doppelstunden auszuweiten, dies ist jedoch aufgrund der schulischen Rahmenbedingungen nicht möglich.

Um diesem Problem entgegenzuwirken, werden die Reflexionsaufgaben verkürzt und den Schülern*innen die Möglichkeit geben, entweder ein Erfolgserlebnis oder eine Herausforderung aus dem Unterricht zu reflektieren. Weil beide Ereignisse sich aus der gelernten Erfahrung leider zeitlich nicht umsetzen lassen, und Hausübungen in diesem Setting leider nicht erlaubt. Die Aufgabe im Lerntagebuch wird somit folgendermaßen modifiziert:

Überschrift des Lerntagebuch-Eintrages: **Wenn-Funktionen**

Reflektiere über einen Moment während des Programmierens oder Vorübungen des Programmierens, indem du ein **Erfolgserlebnis** hattest. Was hat dazu beigetragen, dass du dich kompetent und selbstwirksam gefühlt hast, und wie kannst du dieses Gefühl in zukünftigen Programmieraufgaben nutzen?

Falls du kein Erfolgserlebnis hattest, beschreibe einen Zeitpunkt im Unterricht, indem du dich herausgefordert gefühlt hast. Was war gut und was war schlecht bezüglich der Erfahrung? Welchen Sinn kannst du aus der Situation ziehen? Was hättest du anders machen können? Wie kannst du diese Erfahrung in zukünftigen Programmieraufgaben nutzen?

Abb. 24 Lerntagebuch aus dem Onlinekurs Klasse B

5.2.2 Durchgang 2

Die meisten Schüler*innen beschrieben in den Reflexionen hauptsächlich das Blockly-Spiel, weswegen die Reflexion und Analyse des Spieles Lightbot nur auf die Beobachtungen der Lehrperson zurückzuführen sind. Grundsätzlich konnte beobachtet werden, dass alle Schüler*innen in verschiedenen Geschwindigkeiten zumindest drei Level geschafft haben und auch Schleifen anwenden konnten. Zwei Schüler*innen waren deprimiert und wollten auf dem schnellen und einfachen Weg die Lösung von der Lehrperson erfragen, was jedoch abgeblockt wurde. Sie wurden aufgefordert, es selbst zu versuchen, da sie schon so weit fortgeschritten waren und sie dies auch schaffen würden. Einer dieser Schüler schrieb anschließend in der Reflexion:

„Ich wollte das Level einfach soooo gerne schaffen, das ich von allen Malen, wo ich gescheitert bin, abgesehen habe und mich der Herausforderung gewendet habe. Und wenn man es einmal kapiert

hat, wie man diesen Roboter programmieren muss, dann ist es echt einfach! Alles in allem hab ich vieles für zukünftige Programmierungsaufgaben gelernt und bin (glaub ich zumindest) für die nächste Herausforderung bereit! (Schüler)“

Dies lässt darauf schließen, dass es die richtige Intervention war den Schüler zu ermutigen selbst weiter daran zu arbeiten und nicht gleich aufzugeben. Nach den Einflussfaktoren nach Bandura (1997) ist dies die verbale Überzeugung, die hier Früchte getragen hat.

Eine gute Intervention war die Aufforderung der Schüler*innen beim Erarbeiten der Wenn-Funktionen ihr eigenes Beispiel zu überlegen. Zu einem wurde ein gutes Beispiel genannt und zum anderen erhalten die Schüler*innen somit auch die Möglichkeit ihr Wissen zu präsentieren und Lob dafür zu ernten.

Bei Blockly war die Integration von Peer-Sitzungen förderlich, weil hier die Schüler*innen einander ermutigen können und auch sehen, dass auch ihre Mitschüler*innen es schaffen und somit hoffentlich dies ebenfalls auf sich selbst replizieren.

Bei Blockly erwähnten sieben verschiedene Schüler*innen, dass sie es zu Beginn als Herausforderung betrachteten. Doch mit genügend Anstrengung konnten sie es schaffen. Zwei der sieben Schüler*innen sahen dies auch als Antrieb für zukünftige Schwierigkeiten bei Programmieraufgaben.

5.3 Dritte Unterrichtseinheit – Pair Programming

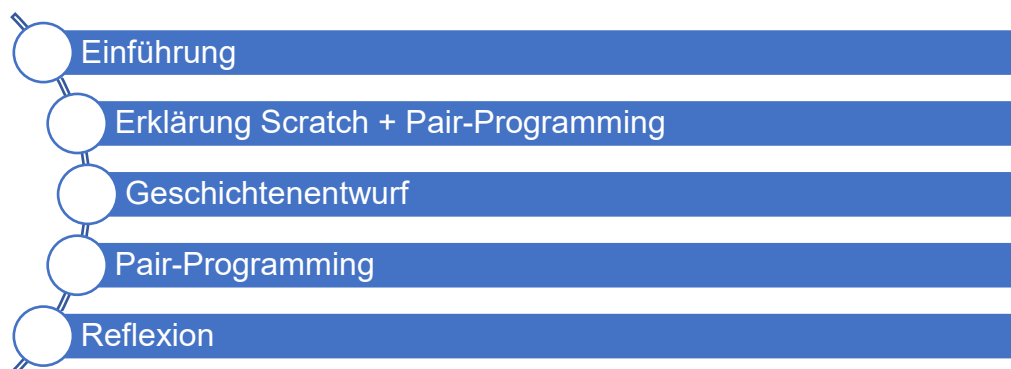


Abb. 25 Übersicht dritte Unterrichtseinheit

5.3.1 Durchgang 1

Die Geschichte der Ada Lovelace war eine subtile Intervention. Direkte Auswirkungen auf die Selbstwirksamkeit konnten durch die Beobachtung, aber auch durch die Reflexion der Schüler*innen, nicht wahrgenommen werden. Jedoch bringt die Geschichte Vorbilder und im weiteren Sinne beobachtbare Erfahrung, welche vor allem die Mädchen hier finden konnten. Diese Faktoren sind laut Tellhed et al. (2022) bei Jungen mehr vorhanden. Somit wurde dem entgegengewirkt und weibliche Vorbilder für Mädchen sichtbar gemacht.

Der anschließende Vortrag zur Erklärung des Pair-Programming und Scratch ist für die Selbstwirksamkeit beim Programmieren nicht direkt entscheidend. Eventuell sehen jedoch wieder vor allem Mädchen den oben genannten Vorbild-Einfluss sowie beobachtbare Erfahrung. Jedoch muss hier erwähnt werden, dass dieser Einfluss nach Kriegler-Kastelic (2018) geringer ist, wenn die erfahrende Person der beobachtenden Person nicht ähnlich ist, wovon hier ausgegangen werden kann. Eine mögliche Intervention wäre eine*n Schüler*in, die aus privater Erfahrung Scratch kennt, mit einzubinden.

In den Reflexionen gingen alle Schüler*innen auf ihre Erfahrungen beim Pair-Programming ein. Zwölf Schüler*innen schilderten trotz der kurzen Zeit von 15 Minuten ein Erfolgserlebnis. Sechs Schüler*innen schilderten von aktiven Meistererfahrungen, eine Schülerin schrieb beispielsweise:

„es war dann aber doch ganz schön kompliziert, dn Code so zu schreiben, dass die katze nicht in einen Stein oder so reinläuft. Deshalb war es ein noch coolerer erfolgserlebnis, als es dann nach längerer zeit funktioniert hat (Schülerin).“

Insgesamt vier Schüler*innen erwähnten in ihren Reflexionen, dass sie zu wenig Zeit hatten. Dies wurde in der Beobachtung ebenfalls betrachtet und analysiert. Diese Schwäche ist eindeutig, kann aber aufgrund der Rahmenbedingungen nicht geändert werden. Spannend für weitere Forschung, wäre ein ähnliches Konzept zu erarbeiten und den Schüler*innen immer die doppelte Zeit zu geben. Dies betrifft vor allem die Übungsphasen.

5.3.2 Durchgang 2

Die Schüler*innen erwähnten in den Reflexionen nichts in Bezug auf den Stundeneinstieg, somit ist es wie im ersten Durchgang beschrieben, schwierig den Einfluss auf die Selbstwirksamkeit festzustellen. Jedoch konnte dieses Mal an der Mimik der Schüler*innen erkannt werden, dass sie in Bezug auf das Kommende motiviert waren. Dies ist zumindest eine gute Basis.

Die Intervention, dass die Schüler*innen helfen Scratch zu erklären, hat ausgesprochen gut funktioniert. Da doch Schüler*innen bereits privat Erfahrungen gesammelt haben oder auch gut kombinieren und transferieren können. Dies könnte ausgeweitet werden, indem vorher gefragt wird, ob sich jemand Erfahrung in Scratch hat. Diese*n Schüler*in bedient den Computer, währenddessen die Lehrperson am projizierten Bild erklärt.

Wieder als herausragend hat sich die Einteilung nach den erreichten Punkten im Fragebogen erwiesen. Die Schüler*innen arbeiteten gut zusammen. Vor allem in dieser Klasse ist dies aus Erfahrung schwieriger, wenn die Einteilung bereits gegeben ist. Es konnte gut beobachtet werden, wie produktiv die Schüler*innen somit arbeiteten.

Alle Reflexionen beziehen sich, ausgenommen einer Reflexion, auf das Pair-Programming mit Scratch.

„Ich habe Freude und Stolz gefühlt. Ich werden versuchen mich in den nächsten Stunden genau so sehr zu bemühen wie heute. Ich bin über mein heutiges Erfolgserlebnis sehr glücklich. Außerdem war das Programmieren wirklich sehr spannend, interessant und kreativ.“
(Schülerin)“

Ähnlich wie diese Reflexion drückten drei weitere Schüler*innen ein Gefühl von Stolz aus. Es ist möglich dieses Gefühl von Stolz auch als einen Baustein der Selbstwirksamkeit interpretieren. Wenn Stolz, als Endresultat betrachtet wird, eine Aufgabe geschafft zu haben, ist das der Ideale Grundstein die Selbstwirksamkeit zu steigern. Vor allem mit dem in der Reflexion oben verbundenen zukünftigen Bemühen, dass die Schüler*innen beschrieben.

Zwei Drittel der Schüler*innen schilderten eine positive Erfahrung. Dies zeigt auch, dass sich Scratch als gute Grundlage eignet, aktive Meistererfahrungen zu

sammeln, um die Selbstwirksamkeit im Programmieren zu erhöhen. Aus diesen Gründen wird dieser Teil der Einheit beibehalten und nicht abgeändert.

5.4 Vierte Unterrichtseinheit – Pair Programming



Abb. 26 Übersicht vierte Unterrichtseinheit

5.4.1 Durchgang 1

Die Unterrichtsstunde verlief einwandfrei. Auch das Pair-Programming hat sich bereits in den letzten Stunden bewährt. Die Schüler*innen arbeiteten gut zusammen und alle brachen auch gute Lösungen zustande. Bezogen auf den Aufbau und den Inhalten ist die einzige Verbesserung wieder nur die Zeit. Es ist schade, die Schüler*innen in ihrem strebsamen Machen unterbrechen zu müssen, weil die Einheit wieder fast vorbei ist.

Es haben 19 Schüler*innen eine Reflexion abgegeben. Von diesen 19 Schüler*innen haben 13 Schüler*innen erwähnt, dass die Lehrperson für sie eine Vorbildfunktion beim Programmieren lernen einnimmt. Nur ein*e Schüler*in sprach sich explizit dagegen aus. Dies lässt darauf schließen, dass die Lehrperson für die Schüler*innen eine wichtige Rolle als Vorbildfunktion darstellt. Bezüglich der Frage, ob das Geschlecht Relevanz hat, sind elf Schüler*innen der Meinung, dass dies für sie und ihre Selbstwirksamkeit irrelevant sei. Zwei Schüler*innen gaben allerdings explizit an, dass ihre weibliche Lehrperson für sie einen Unterschied macht.

*„Ja sie ist ein Vorbild für mich wenn es um programmieren geht
dank ihr sehe ich das ich auch programmieren lernen kann und wenn
etwas nicht funktioniert die ruhe behalten muss (Schülerin).“*

Aus der Reflexion dieser Schülerin kann man erkennen, dass hier nach Bandura (1997) stellvertretende Erfahrung stattfindet. Auch nach Tellhed et al. (2022) kann ein Vorbild, mit dem sie sich identifizieren kann, ihre Selbstwirksamkeit steigern. Zwei Schülerinnen von elf Schülerinnen sind kein signifikantes Ergebnis, weswegen dies nicht allgemein gesehen werden kann. Vor allem, weil sich weitere Schüler*innen

explizit dagegen aussprechen, dass das Geschlecht hier eine Rolle spielt. Jedoch wenn es nur für manche wirkt, wirkt es trotzdem. Und es zeigt sich, wie Weinhardt (2017) bereits erwähnt hat, dass Rollenbilder vor allem für Mädchen oft fehlen.

An dieser Einheit wäre es eine Möglichkeit beim Pair-Programming an dem Modus der Zeit (länger als fünf Minuten) zu experimentieren. Es wird des Weiteren beibehalten den Schüler*innen noch eine zusätzliche Stunde zu geben und sie somit ihren kreativen Prozess ausleben zu lassen.

5.4.2 Durchgang 2

Die Beobachtung, wie sich die neuen Teams beim Arbeiten eindeutig schwerer getan haben, war äußerst spannend. Diese Erkenntnis, wie die Teams besser zusammengearbeitet haben, als sie noch einen ähnlichen Score bei der Selbstwirksamkeit im Programmieren hatten, ist sehr aufschlussreich. Diese Erkenntnis unterstützt das Ergebnis von Palmer und Rao (2019), dass die Ähnlichkeit der Selbstwirksamkeit einen signifikanten Einfluss auf den Lernerfolg und Zufriedenheit hat. Dies führte andersrum dazu, dass ein Schüler seinen Teamkollegen in der Reflexion als „dumm“ bezeichnet hat und sich über die mangelnde Leistung beschwerte. Ein Schüler ist nicht aussagekräftig für die Allgemeinheit, zeigt jedoch gut, wie schon erwähnt, das Ergebnis von Palmer und Rao (2019) in dieser Reflexion zusätzlich zu der Beobachtung im Unterricht. Ein neu zusammengewürfeltes Team, welches sich jedoch im Score ähnlicher war, beschrieb ein Erfolgserlebnis, von dem sie positiv für die Zukunft zehren können. Insgesamt wurden von den Schüler*innen aus elf Reflexionen die Kategorie positiver Einfluss auf die Selbstwirksamkeit (mehr als die Hälfte hatte eine erfolgreiche Meistererfahrung) siebenmal erwähnt. Hier würde sich für weitere Forschung als aufschlussreich erweisen, ob es sich negativ auswirkt, wenn die Partner*innen unterschiedlicher sind. Sowie ob im weiteren Sinne ein größerer Lernerfolg und Zufriedenheit erreicht wird, wenn sie stattdessen in Einzelarbeit weiterarbeiten.

Bezüglich der Veränderung des 7-Minuten-Modus wurde keine Veränderung beobachtet, sowie auch nicht von den Schüler*innen in den Reflexionen erwähnt. Deswegen würde ich auf den fünfminütigen Wechsel wieder zurückkommen, da dadurch der Unterricht aktiver und dadurch, meiner Meinung nach, aktivierender auf die Schüler*innen wirkt.

Bei den Ergebnissen bezüglich der Aufgabe „Denke darüber nach, ob deine Lehrperson ein Vorbild für dich ist, wenn es darum geht, Programmieren zu lernen, ...“ kam es zu folgenden Antworten: Mehr als die Hälfte, genau sieben Schüler*innen sehen die Lehrperson als Vorbild beim Programmieren lernen. Insgesamt wie oben erwähnt, gab es elf ausgefüllte Reflexionen. Eine Reflexion war widersprüchlich, da die Schüler*in schrieb „1. Ich denke nicht das meine lehrerin mein vorbild ist. Aber ich würde mich gerne so gut in informatik auskennen wie sie (Schülerin).“ und wurde deswegen einmal zu der Kategorie Vorbild, sowie kein Vorbild gezählt. Somit gab es noch ein*e Schüler*in, die hierzu keine Angabe machte. Wie beim ersten Durchgang zeigte sich somit, dass die Lehrperson für den Großteil auch eine Vorbildfunktion beim Programmieren lernen hat. Ebenso redundant mit dem ersten Durchgang ist für zwei Drittel der Schüler*innen das Geschlecht irrelevant. Nur ein Mädchen schrieb, dass sie es gut findet zu sehen, dass auch eine Frau diesen „Beruf mit Programmieren“ (Wortlaut der Schülerin) ausüben kann. In der Studie von Kriegler-Kastelic (2018) wird erörtert, dass der Einfluss der stellvertretenden Erfahrung auf die Selbstwirksamkeit höher ist, umso ähnlicher die Person der beobachteten Person ist. Aus diesen Ergebnissen kann die Schlussfolgerung gezogen werden, dass das gleiche Geschlecht (in diesem Fall weiblich) der Lehrperson allein nicht reicht.

Zusammenfassend würde deswegen die Veränderung für eventuelle zukünftige Einheiten und das schlussendliche Konzept folgendermaßen sein: Zurück zu dem fünfminütigen Wechsel, jedoch individuell auf die Klasse und Umstände anpassbar. Sowie die Empfehlung, wenn sich Teams aufgrund Umstände auflösen müssen, entweder Schüler*innen mit ähnlicher Selbstwirksamkeit im Programmieren zusammengeben oder sie allein am Projekt weiterarbeiten lassen.

5.5 Fünfte Unterrichtseinheit – Präsentation und Feedback



Abb. 27 Übersicht fünfte Unterrichtseinheit

5.5.1 Durchgang 1

Die Vorgehensweise, ein gutes Feedback gemeinsam mit Schüler*innen zu erarbeiten, wird nicht verändert, da somit die Schüler*innen gleich abgeholt und auf die Stunde eingestimmt wurden. Der Großteil der Schüler*innen waren der Beobachtung nach überfordert, gutes Feedback zu geben, weswegen der Punkt, dass das Feedback im Dialog stattfinden muss, nicht aufgenommen wird in die zukünftige Planung. Den Auftrag, mindestens eine Stärke zu nennen, erachte ich als sinnvoll. Damit können die Schüler*innen versuchen, sich mehr mit der Geschichte beziehungsweise dem Code der Mitschüler*innen auseinander zu setzen.

Die Veränderung der nächsten Einheit hat den Sinn eine tiefgreifendere Reflexion zu erreichen und die Schüler*innen somit zu motivieren, sich mehr mit dem Thema auseinanderzusetzen. Es wird verändert, dass gemeinsam erarbeitet wird, wie ein Feedback anhand von einem Beispiel aussehen kann, bevor die Schüler*innen einander Feedback geben. Das Chaos sorgte für Verwirrung, weswegen nächste Stunde auf Vierer-Gruppen gesetzt wird, welche einander Feedback geben.

Aus den Reflexionen, welche insgesamt wegen der Umstände von sechs Schüler*innen stammten, erwähnten zwei Schüler*innen, dass sie vorher Angst hatten, schlechtes Feedback zu erhalten. Diese von Angst geprägte Umgebung widerspricht der von Allaire-Duquette et al. (2022) betonten wichtigen angstfreien Umgebung. Mit der Veränderung, mit den Schüler*innen im Vorhinein ein gemeinsames Feedback zu erarbeiten, kann dies hoffentlich den Schüler*innen helfen die Angst zu nehmen.

Auswirkungen der Einheit auf die Selbstwirksamkeit beim Programmieren konnte auch bei vier von sechs Reflexionen analysiert werden. Eine Schülerin schrieb beispielsweise „Und da das Feedback gut war wusste ich das meine Geschichte gut ist und ich programmieren kann (Schülerin).“ Diese Schülerin bezog sich auf das Feedback wie auch zwei andere Schüler*innen. Die restlichen drei Schüler*innen reflektieren positive Veränderungen direkt beim Erstellen des Scratch-Programmes beziehungsweise Veränderungen, die sie zwischen den Feedbacks durchgeführt haben.

5.5.2 Durchgang 2

Die Intervention, mit den Schüler*innen gemeinsam ein Feedback zu erstellen, zeigte sich als sehr erfolgreich. Die Schüler*innen gaben anschließend mehr förderliches und genaueres Feedback. Es ist eigentlich auch eine gute Möglichkeit,

den Schüler*innen nochmal die Geschichte von Ada Lovelace zu präsentieren und somit den Mädchen zu zeigen, dass nicht nur Frauen auch programmieren können, sondern eine sogar zu den Pionierinnen des Programmierens zählt.

Der neue Feedback Modus brachte auch mehr Fokus auf den eigentlichen Aspekt, und zwar auf das Feedback. Das Chaos des letzten Durchgangs konnte somit behoben werden und die Intervention zeigte sich als erfolgreich. Eine mögliche zukünftige Intervention wäre das Feedback wegzulassen, da Schüler*innen in der Reflexion Angstzustände vor dem Feedback beschrieben hatten. Stattdessen wäre der Plan, dass sich die Lehrperson von jeder Gruppe einmal die Geschichte zeigen lässt und Selbstwirksamkeit förderndes Feedback gibt. Dies liegt auch am Ermessen der Lehrperson, wie gut solche Feedbackrunden mit der Klasse funktionieren.

In dieser Klasse hat das Feedback in diesem Modus gut funktioniert, vor allem bei der Betrachtung der Reflexionen. Sechzehn der zwanzig Schüler*innen beschrieben, dass sie das erhaltene Feedback in ihrer Selbstwirksamkeit beim Programmieren gestärkt hat. Zwei Schüler*innen konnten keine Veränderung bemerken. Die restlichen zwei Schüler*innen haben kein Feedback erhalten, da sie ihre Geschichte nicht mehr fanden und somit nichts präsentieren konnten. Dieses eindeutige Ergebnis spricht für ein Beibehalten der Feedbackrunden.

6. Ergebnisse der Fragebögen

Schlussendlich nahmen vierzig Schüler*innen zwischen zwölf und dreizehn Jahren an der Forschung teil. Bei der ersten Erhebung der Selbstwirksamkeit im Programmieren am Anfang der ersten Unterrichtseinheit füllten zwanzig Schülerinnen und zwanzig Schüler den Fragebogen aus. Niemand der Schüler*innen identifizierte sich als divers. Bei der zweiten Erhebung fehlten fünf Schülerinnen, weswegen hier die Werte der restlichen 35 Schüler*innen ausgewertet werden.

6.1 Fragebogen 1

Bei dem ersten Fragebogen zeigte sich, dass die erreichten Werte der Selbstwirksamkeit im Programmieren der Schülerinnen deutlich unter denjenigen der Schüler liegen. Wie in Abbildung 13 erkennbar liegt die graue Linie der sortierten Werte der Schülerinnen unter der blauen Linie der Werte der Schüler. Der höchste erreichte Wert eines Schülers war 3,7. Den niedrigsten Wert hatte eine Schülerin mit 1,7. Der Durchschnitt der vierzig Schüler*innen lag bei 2,82.

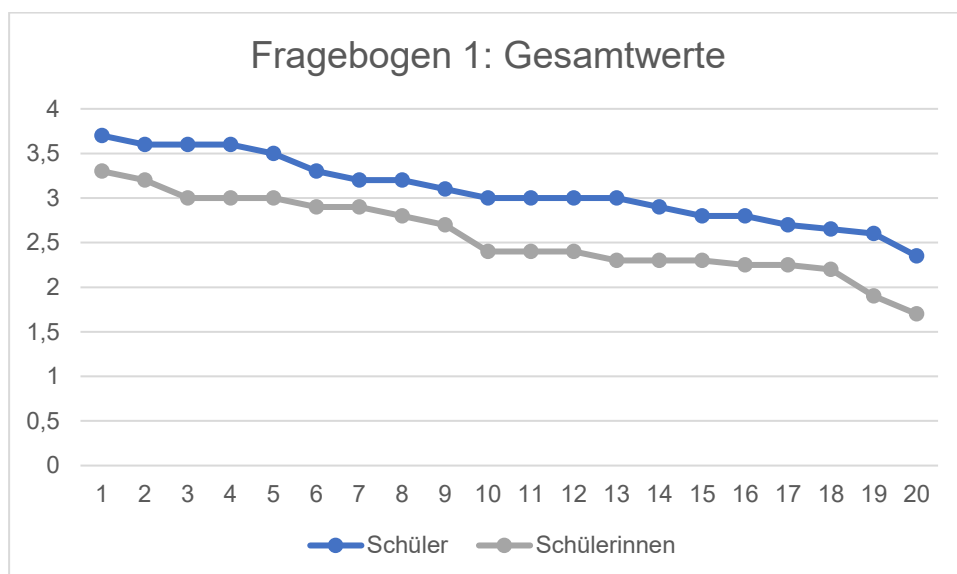


Abb. 28 Fragebogen 1 Gesamtwerte

Bei der ersten Erhebung zur Messung der Selbstwirksamkeit im Programmieren erreichten die zwanzig männlichen Schüler Werte von 2,35 bis zu 3,7 mit einem Mittelwert von 3,08. Die Schüler werden im folgenden Diagramm mit blauer Farbe dargestellt.

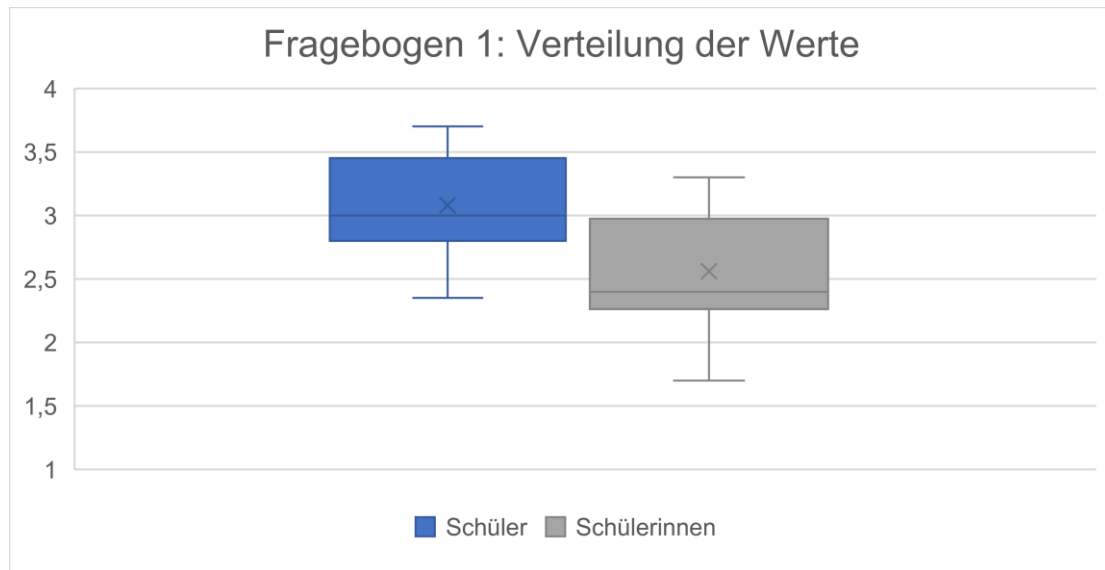


Abb. 29 Fragebogen 1 Verteilung der Werte

Im Vergleich erzielten die zwanzig Schülerinnen Werte im Bereich von 1,7 bis zu einem Höchstwert von 3,3. Der Mittelwert mit 2,56 liegt somit um 0,52 unter dem der männlichen Schüler.

Der durchschnittliche erreichte Wert der Selbstwirksamkeit im Programmieren der Schülerinnen ($M=2,56$) war beim ersten Fragebogen vor der Durchführung des Unterrichtskonzeptes geringer als der der Schüler ($M=3,08$). Der Unterschied war höchst signifikant: $p = 0,00013579$. Die Schüler haben sich somit zu Beginn höchst signifikant höher in der Selbstwirksamkeit im Programmieren eingeschätzt.

6.2 Fragebogen 2

Unter Betrachtung des zweiten Fragebogens nach Durchführung des Inklusiven Unterrichtskonzeptes hat sich der Unterschied zwischen den Geschlechtern vermindert. Die Schüler*innen füllten den abschließenden Fragebogen am Ende der fünften und letzten Unterrichtseinheit aus. In der zweiten Erhebung wurden insgesamt fünfzehn Fragebögen ausgewertet, da fünf Schülerinnen krankheitsbedingt fehlten. Von den zwanzig männlichen Schülern füllten wieder alle den zweiten Fragebogen aus.

Insgesamt schätzten sich die gesamten Schüler*innen mit einem durchschnittlichen Wert von 3,264 ein. Die Werte variierten von 3,8 bis 2,45 Punkten.

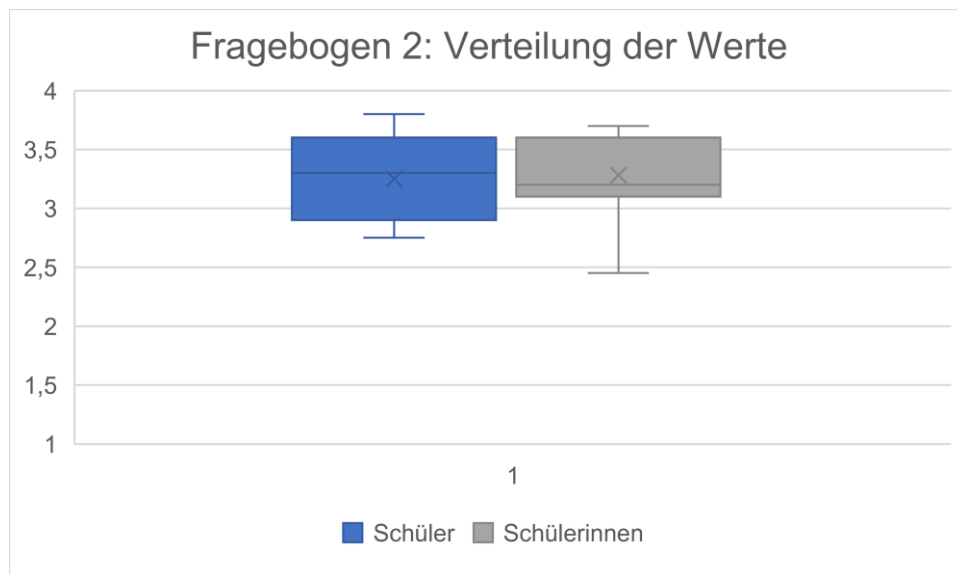


Abb. 30 Fragebogen 2 Verteilung der Werte

Die Schülerinnen schätzten sich bei der Selbstwirksamkeit im Programmieren mit einem durchschnittlichen Wert von 3,28 ein. Der Wert nach Durchführung des Konzeptes stieg im Vergleich zur ersten Erhebung um 18 %. Der niedrigste Gesamtwert betrug 2,45 Punkte und der Höchstwert von 3,7 wurde von zwei Schülerinnen erreicht.

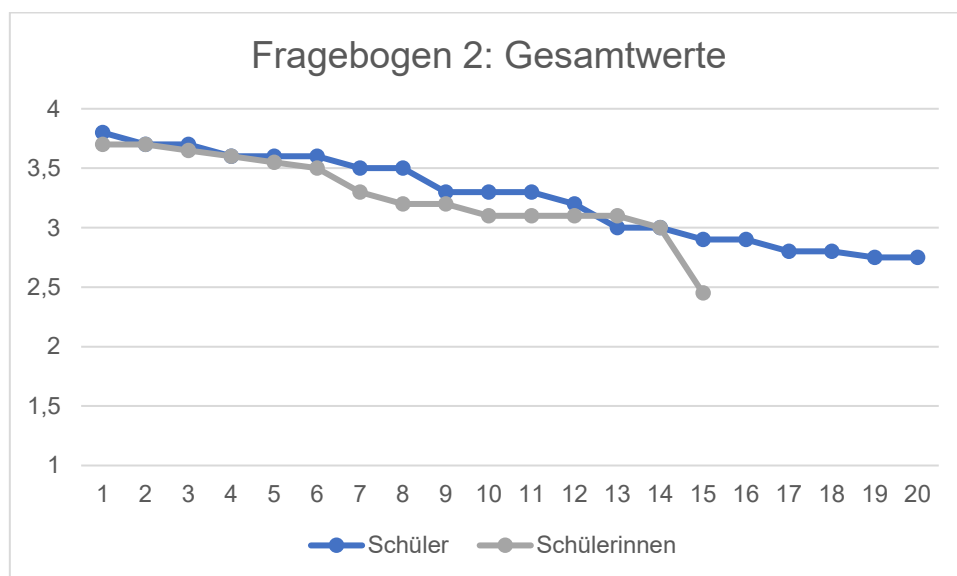


Abb. 31 Fragebogen 2 Gesamtwerte

Im Gegensatz zu den männlichen Schülern stieg der durchschnittliche Wert bei den Schülerinnen von 3,08 auf 3,25, insgesamt um 5,5 %. Der höchste Wert ist 3,8 und der niedrigste Wert liegt bei 2,75 Punkten. Die Spannweite der Werte variiert zwischen den Geschlechtern und ist bei Schülerinnen geringer im Vergleich zu Schülern.

Der durchschnittliche erreichte Wert der Selbstwirksamkeit im Programmieren der Schülerinnen (M=3,28) war beim zweiten Fragebogen nach der Durchführung des Unterrichtskonzeptes ähnlich wie der der Schüler (M=3,25). Dadurch lässt sich zwischen den Geschlechtern mit einer Zufallswahrscheinlichkeit von 40% keinen statistischen Unterschied mehr messen.

6.2.1 Offene Freitextfragen

Die erste Frage war „Inwiefern spielt für dich die Lehrperson eine Rolle bei der Begleitung beim Programmieren-Lernen?“ und wurde von allen 35 Schüler*innen im Freitextfeld beantwortet.

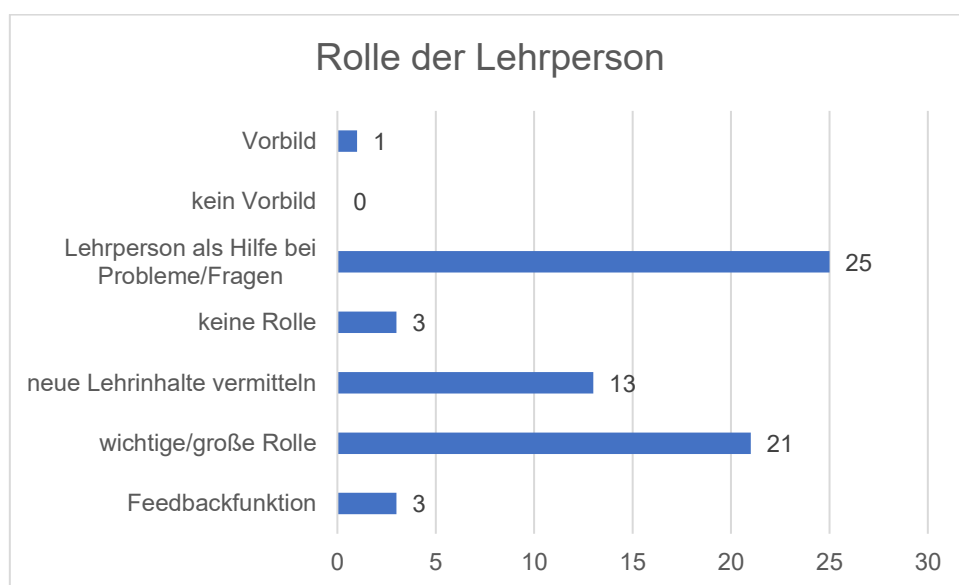


Abb. 32 Auswertung Rolle der Lehrperson

Am häufigsten betrachteten die Schüler*innen die Lehrperson beim Programmieren-Lernen in der Rolle der Ansprechperson, welche bei Problemen oder Fragen Hilfestellungen leisten kann. Insgesamt sahen auch mehr als die Hälfte der Schüler*innen eine bedeutende Rolle der Lehrperson als Begleitung beim Programmieren-Lernen.

Bezogen auf die Vermutung, dass die Lehrperson als Vorbild eine Rolle spielen könnte, sind vor allem die Kategorien Vorbild und kein Vorbild interessant. Nur ein*e Schüler*in hat explizit erwähnt, dass die Lehrperson als Vorbild beim Programmieren-Lernen fungiert.

*"Sie ist für mich Vorbild, weil sie sehr gut programmieren kann,
und weil sie Zeit, dass in der heutigen Gesellschaft auch „eine Frau“ mit
Computern umgehen kann (Schüler*in)."*

Allerdings hat sich auch kein*e Schüler*in explizit die Rolle der Lehrperson als Vorbild als unwichtig dargestellt.

Interessant ist bezüglich der Einflussfaktoren der Selbstwirksamkeit auch der Code Feedbackfunktion, dem drei Aussagen von Schüler*innen zugeordnet worden sind. Ein*e Schüler*in schrieb:

*"Ebenso kann sie mich außbessern und weitere dinge
verbessern. Somit kann ein schüler immer besser und besser werden.
Dadurch habe ich zuvericht, mal gut programieren zu können
(Schüler*in)."*

Hervorzuheben ist auch noch, dass für drei Schüler*innen die Lehrperson beim Programmieren-Lernen als nicht relevant gilt. Die Schüler*innen schrieben bei ihren Antworten, dass ein*e Schüler*in bereits die Basics zum Programmieren kannten, ein*e Andere*r sich mehr darauf konzentrierten die Probleme im Team zu lösen oder der*die dritte Schüler*in immer selbstständig an alle Aufgaben herangegangen war.

Die zweite und letzte Frage des abschließendes Fragebogens war folgendermaßen: Hast du während der letzten Unterrichtsstunden eine Veränderung deines Vertrauens in deine Fähigkeiten beim Programmieren bemerkt? Wenn ja, beschreibe welche Faktoren dazu beigetragen haben?

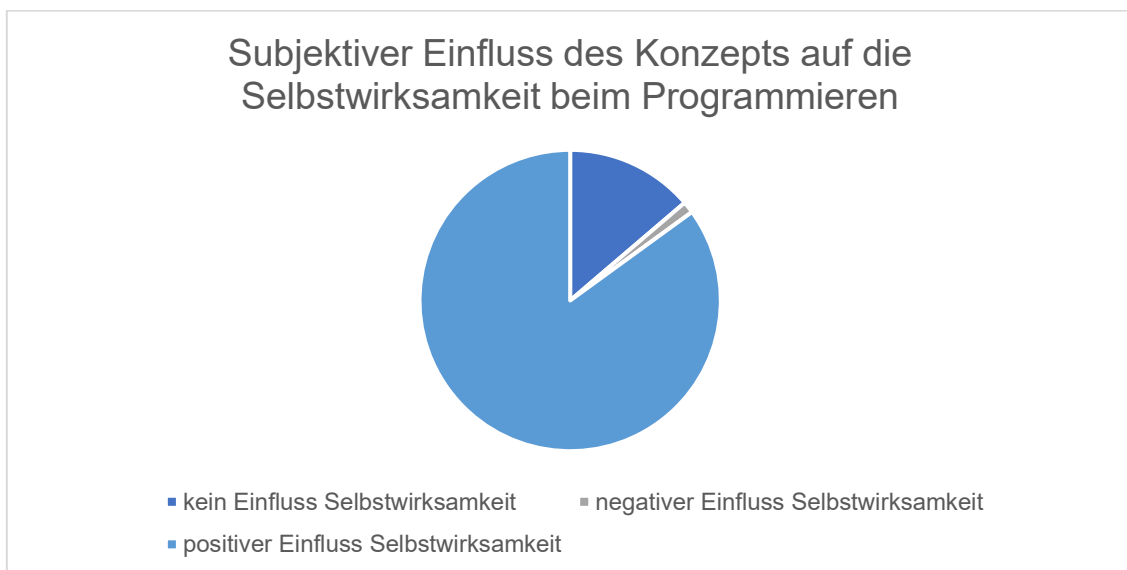


Abb. 33 Auswertung Einfluss Selbstwirksamkeit

Insgesamt haben acht Schüler*innen keinen Einfluss auf ihre Selbstwirksamkeit beim Programmieren durch das Unterrichtskonzept festgestellt. Ein Schüler beschrieb einen negativen Einfluss auf die Selbstwirksamkeit „Nein, weil es mir schwerfällt (Schüler)“. Im Gegensatz beinhalten 68 Segmente beziehungsweise 25 Aussagen und somit die eindeutige Mehrheit eine Beschreibung eines positiven Einflusses auf die Selbstwirksamkeit beim Programmieren.

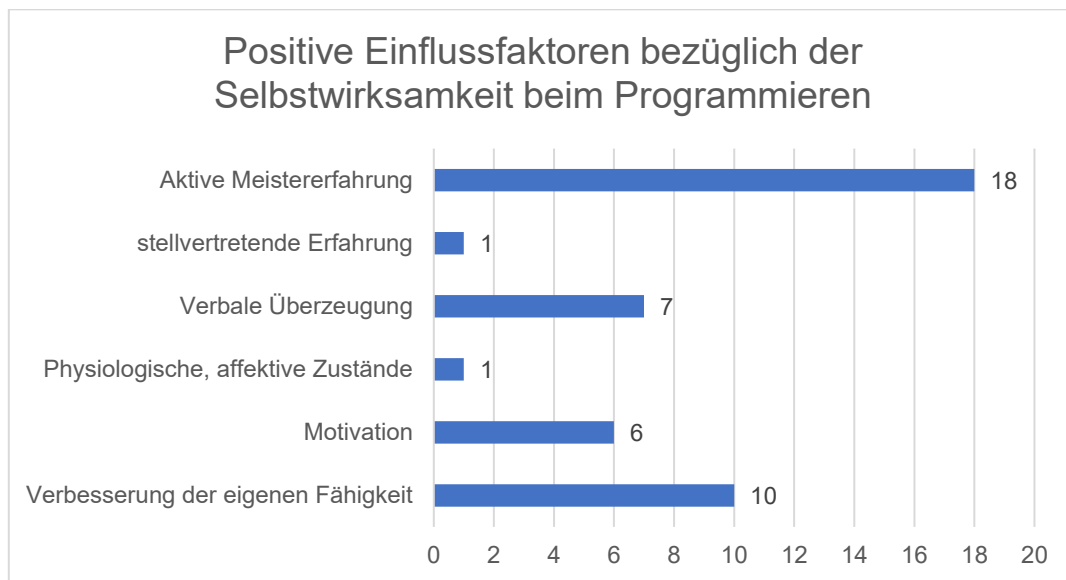


Abb. 34 Auswertung positive Einflussfaktoren

Die am häufigsten genannten Einflussfaktoren waren die Aktive Meistererfahrung, Verbesserung der eigenen Fähigkeit sowie die verbale Überzeugung und Motivation.

Am häufigsten erwähnten die Schüler*innen die aktive Meistererfahrung mit 18 Nennungen. Von den 18 Nennungen bezogen sich zehn Aussagen konkret auf eine aktive Meistererfahrung beim Pair-Programming in Scratch.

Insgesamt handelt es sich bei zehn Aussagen um die Verbesserung der eigenen Fähigkeiten beim Programmieren. Teilweise bezogen sich die Schüler*innen direkt auf die Verbesserung in Scratch. Eine Schüler*in erwähnt auch, dass sie nun besser "logisch über Probleme nachdenken\lösen kann (Schüler*in)."

Drei Schüler*innen erwähnten die positive verbale Überzeugung der Mitschüler*innen, dass sie das Lob oder positive Feedback beeinflusst hat. Viermal fand die verbale Überzeugung laut den Schüler*innen-Antworten auch von Seiten der Lehrperson statt.

Unter dem Code „Motivation“ findet man Antworten fünf Schüler*innen in Form der Beschreibung von Spaß beim Programmieren, aber auch im Unterricht. Gleichfalls hat ein*e Schüler*in angemerkt, dass sich ihre Ziele beim Programmieren gesteigert haben.

6.3 Vergleich Fragebogen 1 und 2

In diesem Kapitel wird die Differenz zwischen den ersten und zweiten Fragebögen der 35 Schüler*innen betrachtet, da fünf Schülerinnen beim zweiten Fragebogen fehlten.

Die Verbesserung in dem Score bezüglich der Messung der Selbstwirksamkeit im Programmieren der fünfzehn Schülerinnen war vom ersten Fragebogen ($M=2,68$) auf den zweiten Fragebogen ($M=3,28$) höchst signifikant ($p=0,00007$). Genauso zeigten die Schüler eine höchst signifikante Verbesserung ($p=0,00676$) vom ersten Fragebogen ($M=3,08$) hin zu dem zweiten Fragebogen ($M=3,25$) nach der Durchführung des Unterrichtskonzeptes.

Schüler	Schülerinnen
-0,2	0,05
-0,2	0,1
-0,1	0,2
-0,1	0,2
0	0,2
0	0,3
0	0,5
0,1	0,65
0,1	0,8
0,1	0,8
0,1	0,8
0,2	0,9
0,2	0,9
0,3	0,9
0,3	1,75
0,3	
0,4	
0,4	
0,5	
1	

Tab. 1 Differenz zwischen ersten und zweiten Fragebogen nach Geschlecht

Die größte Veränderung erreichte eine Schülerin mit 1,75 Punkte mehr im Vergleich zwischen den Fragebögen. Bei den Schülern betrug der Höchstwert 1,0

Punkte. Insgesamt liegt der Durchschnitt der Schülerinnen bei 0,6033 Punkten. Währenddessen der Durchschnitt der Schüler mit 0,17 Punkte deutlich darunter lag.

Auffällig sind die beiden Ausreißer-Werte. Eine Schülerin hatte eine Differenz zwischen den ersten und zweiten Fragebogen von 1,75 Punkten und liegt damit deutlich über den Durchschnittswert, genauso wie ihr männlicher Kollege mit einer Differenz von einem Punkt. Genauso interessant sind allerdings auch die vier negativen Veränderungen bei den Schülern. Ebenfalls spannend sind die drei Schüler, bei denen keine Veränderung in der Selbstwirksamkeit im Programmieren stattgefunden hat.

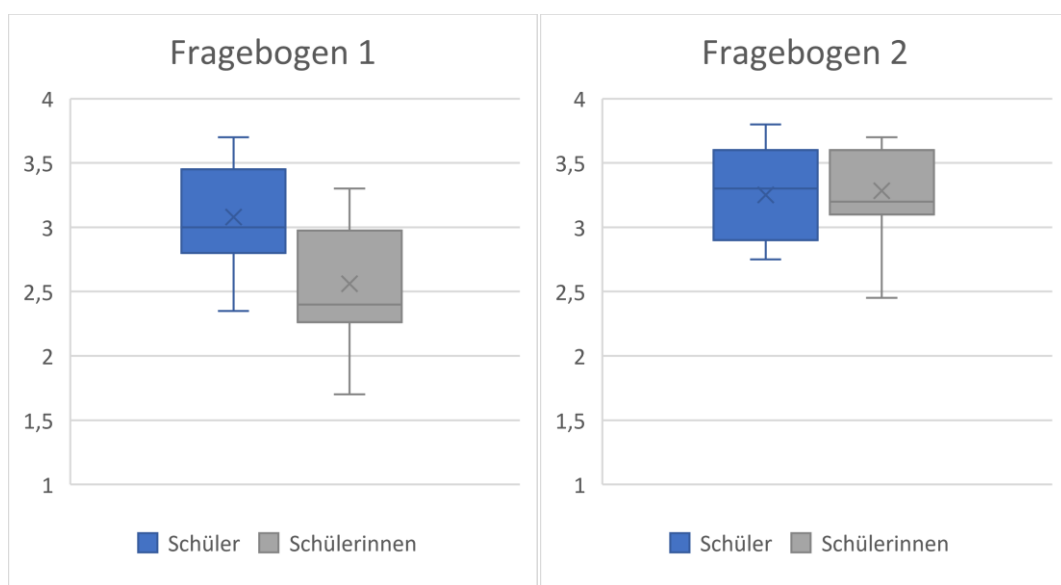


Abb. 35 Vergleich Werte Fragebogen 1 & 2

Betrachtet man Abbildung 35, ist sehr deutlich, dass am Ende des Unterrichtskonzeptes kein Unterschied mehr messbar war. Dies lässt sich auch statistisch mit einer Zufallswahrscheinlichkeit von 40% belegen.

7. Diskussion

7.1 Limitationen

Die zentralen Limitationen dieser Masterarbeit finden sich in den Rahmenbedingungen wieder. Da die Aktionsforschung im Rahmen meines Unterrichtes durchgeführt worden ist, deckt die Forschung zwei Klassen aus derselben Schule mit den identen demografischen Bedingungen ab. Alle Schüler*innen besuchen die Schulform Gymnasium mit dem Realen-Wahlzweig. In diesem Zusammenhang wäre es für weitere Forschung lohnenswert, das Konzept auch in anderen Schulformen, wie in der Mittelschule oder auch in Gymnasialklassen ohne Realzweig zu untersuchen. Diesbezüglich könnte man vor allem in dem Unterrichtsfach „Digitale Grundbildung“, welches in ganz Österreich in der Unterstufe unterrichtet wird, weitere Forschung betreiben.

Die Rahmenbedingungen führten auch zu der Limitation der Methodik der Aktionsforschung mit den unterteilten Aktionsforschungen jeder Unterrichtseinheit mit zwei Zyklen. Da alle Schüler*innen einer Schulstufe den Stoff, abgestimmt auch auf die Jahresverteilung des Unterrichtsfaches Digitale Grundbildung, ungefähr zeitgleich unterrichtet bekommen, konnte das Konzept nicht in einer Klasse durchgeführt, evaluiert und verbessert werden, bevor es in der nächsten Klasse zur Anwendung kam. Deswegen ist es auch nicht möglich mithilfe der Endergebnisse zu diskutieren, ob die Interventionen und Veränderungen im Unterricht positiv oder weniger hilfreich waren. Durch Feiertage und Schulausflüge veränderte sich ebenfalls, welche Klasse als erstes eine Unterrichtseinheit hatte.

Wie schon in der Reflexion und Analyse der Aktionsforschung mehrmals erwähnt, erwies sich als größte Limitation in der Aktionsforschung, die durch die Rahmenbedingungen vorgegebenen 50-Minuten-Einheiten, sowie die zeitliche Begrenzung von fünf Unterrichtseinheiten für dieses Thema. Für weitere Forschung wäre es sehr aufschlussreich, ob sich 100-Minuten-Einheiten mit gleichem Inhalt mit doppelter Zeit als erfolgreicher herausstellen würden. Eventuell könnten damit auch langfristige, tiefergehende positive Veränderungen in der Selbstwirksamkeit im Programmieren erzielt werden.

Als weitere Limitation dieser Masterarbeit kann auch die gewählte Methodik „Beobachtung“ bei der Aktionsforschung angesehen werden. Da hier als generelle Schwächen der Beobachtung nach Altrichter und Posch (2007) die Diffusität,

Vorteilsbehaftigkeit und Flüchtigkeit angesehen werden kann. Zum einen war in dieser Forschung mit Klassengrößen von 25 und 22 Schüler*innen die Aufmerksamkeit meinerseits breit gestreut. Zum anderen ist eine mögliche naheliegende Limitation, dass ich genau die Ziele meiner Arbeit im Speziellen beobachte. Dem entgegenwirken könnte man in zukünftiger Forschung mit einer zusätzlich beobachtenden Person.

Weiterer Forschungsbedarf ergibt sich auch bezüglich der Vorbildwirkung der Lehrperson und dessen Einfluss auf die Programmierselbstwirksamkeit. Um tiefgreifendere Ergebnisse zu erhalten, würde es sich anbieten noch zusätzlich Fokusgruppen oder Interviews mit den Schüler*innen durchzuführen. Dies hätte jedoch den Umfang dieser Arbeit gesprengt.

Des Weiteren wäre es aufschlussreich, ob Schüler*innen, welche von anderen Lehrer*innen unterrichtet werden, insbesondere von männlichen Lehrern, ähnliche Ergebnisse zeigen würden. Dies bezüglich könnte man sich der Frage der Einfluss der Lehrperson als Vorbild auf die Selbstwirksamkeit im Programmieren noch konkreter widmen und hätte auch mögliche Kontrollgruppen.

Bei der Auswertung der Ergebnisse fiel die Normalverteilung weg. Der Grund ist, dass mit 35 teilnehmenden Schüler*innen genügend Daten gesammelt wurden, die Ergebnisse so klar sind und deswegen davon ausgegangen werden kann, dass die Daten repräsentativ sind.

Es ist wichtig zu betonen, dass die Erhebung des letzten Fragebogens, direkt nach der Durchführung des Konzeptes stattgefunden hat. Dementsprechend bilden diese Ergebnisse die Selbstwirksamkeit im Programmieren der Schüler*innen zu diesem speziellen Zeitpunkt ab. Ebenfalls wurde im Weiteren nicht untersucht, ob diese Ergebnisse auch langanhaltend sind, dies würde allerdings ein neues Forschungsfeld eröffnen.

Die gewonnenen Ergebnisse, dass die Schüler vor dem Unterrichtskonzept eine höhere Selbstwirksamkeitserwartung im Programmieren hatten als ihre Mitschülerinnen, wirft die Frage auch, woher diese Einschätzung konkret bei diesen Schülern kommen. Im Laufe der Aktionsforschung stellte sich im Unterricht heraus, dass manche Schüler*innen bereits außerschulische Erfahrungen im Programmieren sammeln konnten. Die Ergebnisse von Tellhed et al. (2022) schließen auf die vorherigen Meistererfahrungen, verbale Überzeugungen, beobachtete Erfahrungen

und Vorbilder der Jungen. Diese Vorerfahrungen wurde in dieser Studie allerdings nicht erhoben.

Ein weiteres Forschungsgebiet würde auch die Weiterführung des Konzeptes im Lehrstoff weiterhin unter dem Schwerpunkt der Steigerung der Selbstwirksamkeit im Programmieren in höheren Klassen darstellen.

7.2 Diskussion der Ergebnisse der Fragebögen

Das höchst signifikante Ergebnis bei der Erhebung der Selbstwirksamkeit im Programmieren vor dem Unterrichtskonzept, zeigt das geschlechtsspezifische Unterschiede bezüglich der Selbstwirksamkeit im Programmieren existieren. Die Schüler schätzen ihre Fähigkeiten im Programmieren eindeutig besser als die Mädchen ein. Dies stimmt mit den Ergebnissen von Sullivan (2015), Allaire-Duquette et al. (2022) und Tellhed et al. (2022) überein. Gleiche Ergebnisse bezüglich der Selbstwirksamkeit im MINT-Bereich konnten Weinhardt (2017) und Girnat et al. (2020) feststellen.

Bei Betrachtung der Ergebnisse der Erhebung der Selbstwirksamkeit im Programmieren im zweiten Durchgang, nach Durchführung des Unterrichtskonzeptes, konnten diese Differenzen aufgelöst werden. Schülerinnen als auch Schüler verbesserten beide höchst signifikant ihre Werte in der Erhebung der Selbstwirksamkeit im Programmieren. Die Schülerinnen konnten sogar sich so weit verbessern, dass kein Unterschied zwischen den Schülerinnen und Schüler messbar war. Das Unterrichtskonzept erzielte somit die Wirkung, dass die Mädchen durch ihre stärkere Verbesserung, das gleiche Niveau bezüglich der Selbstwirksamkeit im Programmieren der Jungs gehoben haben. Somit konnte das Unterrichtskonzept die geschlechtsspezifischen Unterschiede nicht nur minimieren, sondern sogar eliminieren. Die Studie umfasste jedoch nur die Dauer der fünf wöchentlichen Unterrichtseinheiten. Es wurde nicht geprüft, ob dies zu einer langfristigen Steigerung der Selbstwirksamkeit im Programmieren führt. Entscheidend ist diese Frage, da nach Sullivan (2015) die Selbstwirksamkeit in Informatik zu der Geschlechterungleichheit in den Informatik-Studien an den Universitäten führt. Falls sie langfristig halten, könnten das Unterrichtskonzept ein mögliches Instrument bieten, dieser Ungleichheit entgegenzuwirken.

Den Effekt der Steigerung der Selbstwirksamkeit bekamen die Schüler*innen auch subjektiv mit, wenn man nach den Antworten der offenen Freitextfragen geht.

Überraschend ist hier der negative Einfluss auf die Selbstwirksamkeit, welcher sich bei vier Schülern mit einer negativen Veränderung im Score in der Einschätzung der eigenen Fähigkeiten im Programmieren zeigt. Alle vier Schüler hatten beim ersten Durchgang einen Score im oberen Viertel von 3 bis 3,3 (von vier möglichen Punkten). Nun kann diskutiert werden, ob das Unterrichtskonzept einen negativen Einfluss auf sie hatte, da es doch insbesondere darauf bedacht war vor allem die Selbstwirksamkeit der Mädchen zu fördern. Es könnte auch sein, dass die Schüler sich überschätzt haben und durch das Lernen des Programmierens erkannten, dass ihre Einschätzung nicht mit ihren eigentlichen Leistungen korrelierte.

Nach der qualitativen Inhaltsanalyse der ersten Freitextfrage bezüglich der Einflussfaktoren der Selbstwirksamkeit ist das Ergebnis sehr überschneidend mit den Studien von Tellhed et al. (2022) und Allaire-Duquette et al. (2022), dass vor allem die aktive Meistererfahrung den entscheidenden Faktor spielt. Die Schüler*innen nannten ebenfalls mit zehn als auch sieben Nennungen die Verbesserung der eigenen Fähigkeit als auch verbale Überzeugung für die Steigerung der Selbstwirksamkeit im Programmieren entscheidend. Dies zeigt zu einem, wie wichtig es ist, Schüler*innen die Möglichkeit zu geben in das Programmieren reinzuschnuppern, als auch den Unterricht methodisch aufzubauen. Aber auch das Feedback, sei es von Mitschüler*innen oder der Lehrperson, spielt eine Rolle in der Steigerung der Selbstwirksamkeit im Programmieren.

Für die Autorin (mich) überraschend war vor allem nach der Reflexion in der vierten Unterrichtseinheit, dass die Schüler*innen nicht explizit die Rolle der Lehrperson, als die des Vorbildes beim Programmieren-Lernen betrachten. In den Reflexionen haben hingegen bei der konkreten Frage, ob die Lehrperson ein Vorbild beim Programmieren Lernen ist, mehr als die Hälfte die Lehrperson als Vorbild betrachtet. Das Geschlecht erwies sich für jeweils circa zwei Drittel der Schüler*innen als irrelevant. Nach den Antworten der Schüler*innen im zweiten Fragebogen lässt dies darauf schließen, dass das Geschlecht der Lehrperson als Vorbild entgegen der Vermutung der Autorin keine große Rolle spielt. Unter Betrachtung der Studie von Guenaga (2022), welche den Einfluss weiblicher Vorbilder auf die Selbstwirksamkeit untersuchte, stellt dies ein naheliegendes Ergebnis dar. Allerdings reichten diese sechs Einheiten in der Studie von Guenaga (2022) nicht aus, um einen signifikanten positiven Effekt in der Selbstwirksamkeit festzustellen. Vergleicht man dies mit dieser Studie, zeigt dies, dass zwar nicht Vorbilder allein zu einer Erhöhung der

Selbstwirksamkeit, jedoch ein hierfür konzipiertes Unterrichtskonzept es auch in fünf Einheiten erreichen kann. Ein weiterer möglicher Erklärungsansatz liefert auch Kriegler-Kastelic (2018). Hier beschreibt die Autorin, dass der Einfluss der stellvertretenden Erfahrung auf die Selbstwirksamkeit höher ist, umso ähnlicher die Person der beobachteten Person ist. Nun wäre ein weiterer Erklärungsversuch, dass die Lehrperson den Schüler*innen zu wenig ähnlich sei. Genauso, dass das gleiche Geschlecht der Lehrperson für die Erhöhung der Selbstwirksamkeit durch die stellvertretende Erfahrung allein nicht ausreicht.

7.3 Implikationen für zukünftigen Unterricht

Die Ergebnisse aus den Scores der Fragebögen zeigen, dass das Konzept eine höchst signifikante Erhöhung der Selbstwirksamkeit im Programmieren aller Geschlechter in einer 7. Schulstufe in einem Wiener Gymnasium bewirkte. Um die vierte Forschungsfrage zu beantworten, welche Implementierungsstrategien für den Unterricht abgeleitet werden können, vor allem, um im schulischen Umfeld genderspezifische Unterschiede in Bezug auf Gender in der Informatik zu reduzieren, ist eine genauere Betrachtung der Unterrichtseinheiten notwendig. Das Konzept umfasste zusammenfassend vier große Schwerpunkte, das Game-Based-Learning, Pair-Programming in Scratch sowie auf der Reflexion als auch dem Feedback.

Das Game-Based-Learning mit den Spielen „Lightbot“ und „Blockly“ eröffnete den Schüler*innen die Möglichkeiten, dass neu Erlernte sofort umzusetzen mit einer visuellen Betrachtung der Befehle. Bei der Beobachtung der Autorin (meinerseits) konnte vor allem die Motivation der Schüler*innen betrachtet werden. Dies eröffnete möglicherweise einen guten Grundstein für das nach Morales-Navarro et al. (2023) notwendige „growth mindset“, welches signifikant mit den Selbstwirksamkeitsüberzeugungen der Schüler*innen im Bereich des Lernen in der Informatik einhergeht. Mit dem Game-Based-Learning Ansatz bekamen die Schüler*innen auch die Möglichkeit erste Meistererfahrungen zu sammeln. Erfolgreich war auch die Wahl der Spiele „Lightbot“ und „Blockly“, da Bandura (1997) betonte, dass es sich um keinen einfachen Erfolg handeln sollte, sondern mit einer Herausforderung verknüpft werden muss. Genau dies beschrieben die Schüler*innen in ihren Reflexionen der Einheiten. Kritisch betrachtet ist es mit den vorhandenen Daten nicht möglich konkrete Rückschlüsse der Steigerung der Selbstwirksamkeit im Programmieren, vor allem der Schülerinnen, direkt mit den Game-Based-Learning

Ansatz der ersten zwei Unterrichtseinheiten zu verknüpfen. Allerdings stellt diese Methode einen großen Teil des erfolgreichen Konzeptes dar.

Den zweiten großen Teil des Konzeptes bildete das Pair-Programming in Scratch. Für die Autorin (mich) war die größte Überraschung, wie die Schüler*innen die Einteilungen nach den Scores der Selbstwirksamkeit akzeptierten, als auch wie gut die Zusammenarbeit anschließend zwischen den Schüler*innen in den Paaren funktionierte. Es kamen vereinzelt Fragen, ob man nicht doch mit den Freund*innen zusammenarbeiten konnte, doch nach kurzen Erklärungen, wie die Paare zustande kamen, akzeptierten die Schüler*innen sofort die vorher getroffenen Einteilungen. Bezüglich der Zusammenarbeit stellte sich als aufschlussreich die Erfahrung aus der vierten Unterrichtseinheit heraus, als manche Paarung neu zusammengewürfelt worden waren und man beobachten konnte, wie sie im Gegensatz zu der vorherigen Einheit mit einer in der Selbstwirksamkeit ähnliche*n Partner*in besser zusammenarbeiten konnten. Dies stimmt mit den Ergebnissen von Palmer und Rao (2019) überein, dass Ähnlichkeit der Selbstwirksamkeit einen signifikanten Einfluss auf den Lernerfolg und Zufriedenheit hat. In diesem Sinne würde sich die Möglichkeit für weitere Forschung bilden, ob dies einen so signifikanten Einfluss hat, dass das Pair-Programming höhere positive Auswirkungen als Einzelarbeit bietet. Die Schüler*innen beschreiben in den Reflexionen der dritten Unterrichtseinheit mehrheitlich (immer mehr als zwei Drittel der anwesenden Schüler*innen) positive Meistererfahrungen, welche sich nach Kriegler-Kastelic (2018) positiv auf die Selbstwirksamkeit auswirken. Diese signifikante Verbesserung der Selbstwirksamkeit beim Programmieren durch das Pair-Programming stellten auch Tan et al. (2023) und Wei et al. (2021) fest. Auf Grundlage dieser Studien, der Beobachtung als auch den Reflexionen der Schüler*innen wird davon ausgegangen, dass vor allem das Pair-Programming einen Großteil zu der Steigerung der Selbstwirksamkeit im Programmieren innerhalb dieses Konzepts beigetragen hat und dadurch für die Implementierung in den Programmierunterricht sehr empfehlenswert ist.

Bei dem Game-Base-Learning und dem Pair-Programming in Scratch wurde nun häufig die aktive Meistererfahrung genannt. Dies zeigt sich auch bei den offenen Fragen des zweiten Fragebogens. Grundsätzlich kann man aus den Antworten der offenen Fragen bezüglich der Einflussfaktoren entnehmen, dass die Schüler*innen auch vor allem die aktive Meistererfahrung mit 18 Erwähnungen als den Schlüsselfaktor zur Steigerung der Selbstwirksamkeit im Programmieren

wahrnahmen. Die Ableitung dessen ist die Bedeutung für die Selbstwirksamkeit im Programmieren im zukünftigen Programmierunterricht, der den Schüler*innen möglichst viel Raum bieten kann, um viele positive, herausfordernde Erfahrungen zu sammeln. Eine nun erprobte Möglichkeit stellt hierfür das Game-Based-Learning mit „Lightbot“ sowie „Blockly“ und das Pair-Programming in Scratch dar.

Die letzte Einheit des Unterrichtskonzeptes verfolgte in erster Linie den Aspekt des Feedbacks. In den Reflexionen berichteten Schüler*innen über Angstzustände vor dem Feedback der Mitschüler*innen. Nach Bandura (1997) hemmen solche negativen Stressreaktionen die Selbstwirksamkeitserwartung. Dies würde dafürsprechen, zukünftig das Feedback von den Peers wegzugeben und nicht als Empfehlung für zukünftigen Unterricht auszusprechen. Auf der anderen Seite beschrieben 20 von 26 Schüler*innen einen positiven Effekt durch das Feedback ihrer Mitschüler*innen. Der Beobachtung nach tendierten die Schüler*innen dazu die Kreativität bezüglich des Designs zu loben und nicht den Code. Wie Bandura (1997) erwähnt reicht positives Feedback alleine nicht aus. Es müssen die persönlichen Fähigkeiten hervorgehoben werden oder die Verbesserung auf die eigenen Anstrengungen zurückgeführt werden. Dies stellt hohe Anforderungen an die Feedback gebenden Schüler*innen dar. Vor allem, wenn man bedenkt, dass nach Kriegler-Kastelic (2018) dies sogar negativ auf die Selbstwirksamkeit auswirken kann, wenn es als unglaublich eingeschätzt und somit als demotivierend wahrgenommen wird. Diese Punkte machen es schwierig sich für die Implementierung der Feedbackrunde in den Programmierunterricht auszusprechen. Schlussendlich wird es wichtig sein, dass die Lehrperson die Klasse kennt und einschätzen kann, wie gut sie effektvolles Feedback geben können. Eine Möglichkeit, wie man diese möglichen negativen Effekte reduzieren könnte wäre die Feedbackrunde zu streichen und stattdessen die Lehrperson allen Schüler*innen ein individuelles Feedback gibt. Dadurch könnte die Lehrperson auch mehr Fokus darauflegen, dass das Feedback die Selbstwirksamkeit erhöht. Außerdem betont Kriegler-Kastelic (2018), dass das Feedback die größte Wirkung erzielt, wenn die Feedback gebende Person über Expert*innenwissen verfügt und als vertrauens- und glaubwürdig eingeschätzt wird.

Ein fixer Bestandteil aller Unterrichtseinheiten war die Reflexion in Form eines Lerntagebuches. Verfolgt man den Ansatz von Bräuer (2016) ist das Lerntagebuch eine gute Möglichkeit seine Lernfortschritte besser zu verstehen und zu reflektieren. Bestätigt hat sich auch in dieser Forschung die Erkenntnis von Bräuer (2016), dass bei

unzureichender Anleitung Herausforderungen auftreten können. Weswegen die Implementation des Reflexionskreislauf nach Gibbs (1988, zitiert nach (Esslinger-Hinz & Sliwka, 2011) sich als gute Intervention erwiesen hat, da die Schüler*innen nun eine genauere Anleitung hatten, der sie folgen konnten.

Auch wenn in diesem Kapitel nun hauptsächlich auf die großen Implementierungsthemen eingegangen worden ist, sollte nicht auf die kleinen Interventionen während der Forschung vergessen werden. Da die Schüler*innen in den Reflexionen, als auch bei den offenen Fragen nicht auf diese eingegangen sind, lässt es sich nicht konkret sagen, ob diese nun einen wichtigen Beitrag geleistet haben oder nicht. Zusammenfassend waren kleine Interventionen während der Durchführung des Unterrichtskonzeptes das Aufmerksam machen auf die individuelle Steigerung, Peer-Feedback-Sitzungen (auch außerhalb der letzten Einheit), Unterrichtsbeispiel mit Ada Lovelace, Beispiel der Scratch-Geschichte von Schüler*innen vergangener Jahre und das individuelle Feedback der Lehrperson während des Unterrichts. All diese Interventionen (ausgenommen das Unterrichtsbeispiel mit Ada Lovelace) können nach Bandura (1997) und Kriegler-Kastelic (2018) die Selbstwirksamkeit beeinflussen und somit auch zum erfolgreichen Ergebnis dieser Arbeit beigetragen haben können. Das Aufmerksam machen auf die individuelle Steigerung, Peer-Feedback-Sitzung und das individuelle Feedback der Lehrperson sind verbale Überzeugungen. Diese können richtig eingesetzt einen positiven Einfluss auf die Selbstwirksamkeit im Programmieren haben. Unter dem Einflussfaktor „Stellvertretende Erfahrung“ könnten die Peer-Feedback-Sitzungen als auch die Beispiele älterer Schüler*innen wirksam gewesen sein. Vor allem, weil die Lehrperson (ich) zusätzlich erwähnt hat, welche Beispiele konkret von Schülerinnen stammten. Der Ansatz von Tellhed et al. (2022) sieht auch Vorbilder als entscheidenden Faktor der Selbstwirksamkeit beim Programmieren an, wodurch nach dieser Theorie auch das Unterrichtsbeispiel mit Ada Lovelace eine Wirkung, vor allem auf die Mädchen, eventuell erzielen konnte.

8. Fazit

Die vorliegende Studie ging den Fragen nach, ‚Existieren an einem Wiener Gymnasium in der 7. Schulstufe geschlechtsspezifische Unterschiede bezüglich der Selbstwirksamkeit im Programmieren existieren?‘, ‚Zeigt ein spezifisches Unterrichtskonzept mit verschiedenen Unterrichtsdesigns und Ansätzen eine Steigerung der Selbstwirksamkeit im Programmieren bei Schüler*innen der 7. Schulstufe, insbesondere von Mädchen? Kann man mit diesem Unterrichtskonzept die möglichen geschlechtsspezifischen Unterschiede minimieren?‘, ‚Welche Faktoren nennen Schüler*innen als jene, die ihr Vertrauen in ihre Fähigkeiten beim Programmieren beeinflussen?‘, ‚Welche didaktischen Implementierungsstrategien für den Unterricht können aus den Forschungsergebnissen abgeleitet werden, um im schulischen Umfeld genderspezifische Unterschiede in Bezug auf Gender in der Informatik zu reduzieren und somit effektiv der existierenden Genderkluft im Informatikberufssektor entgegenzuwirken?‘. Um diese Fragen zu beantworten, wurde eine Aktionsforschung mit Fragebögen zur Messung des Unterschiedes der Selbstwirksamkeit im Programmieren vor und nach dem Unterrichtskonzept durchgeführt. Der Fragebogen nach dem Unterrichtskonzept wurde mit zwei offenen Freitextfragen, welche mit einer qualitativen Inhaltsanalyse analysiert wurden, ergänzt.

Die Ergebnisse zeigen einen höchst signifikanten Unterschied bei vierzig Schülerinnen und Schüler zwischen 12 und 13 Jahren an einem Wiener Gymnasium in der siebten Schulstufe. Die Schüler erreichten vor dem Unterrichtskonzept einen höchst signifikant höheren Score bei dem Fragebogen zur Einschätzung in der Selbstwirksamkeit im Programmieren im Gegensatz zu den Schülerinnen. Ein spezifisches Unterrichtskonzept mit den Schwerpunkten auf Game-Based-Learning, Pair-Programming in Scratch, Feedback und Reflexion und einer Dauer von fünf Unterrichtseinheiten hat eine höchst signifikante Steigerung in der Selbstwirksamkeit im Programmieren bei den Schülerinnen als auch bei den Schülern bewirkt. Die Mädchen haben eine solch starke Steigerung erreicht, dass statistisch gesehen kein Unterschied in der Selbstwirksamkeit im Programmieren zwischen Jungen und Mädchen mehr zu erkennen ist. Somit konnte das Unterrichtskonzept die Differenzen nicht nur minimieren, sondern eliminieren.

Das Geschlecht der Lehrperson als Vorbildfunktion erwies sich nach der Meinung der Schüler*innen als eher irrelevant für die Steigerung der Selbstwirksamkeit beim Programmieren lernen. Als viel größeren Einfluss hat sich die aktive

Meistererfahrung herausgestellt. Ebenfalls eine höhere Wirksamkeit hat die Verbesserung der eigenen Fähigkeiten, als auch die verbale Überzeugung.

Durch diese Aktionsforschung hat sich somit gezeigt, dass aufgrund eines spezifischen Unterrichtskonzeptes möglich ist, die Selbstwirksamkeit im Programmieren von Schüler*innen der siebten Schulstufe signifikant zu erhöhen und sogar die vorherigen geschlechtsspezifischen Unterschiede aufzulösen. Diese Schüler*innen der 7. Schulstufe maßen dem Geschlecht der Lehrperson, im Gegensatz zur den aktiven Meistererfahrungen, keine hohe Relevanz zu. Die Ansätze Game-Based-Learning, Pair-Programming in Scratch, Feedback und Reflexion können somit als gute didaktische Implementierungsempfehlungen für den Informatikunterricht ausgesprochen werden, um die Selbstwirksamkeit im Programmieren, insbesondere von Mädchen, zu erhöhen und somit die genderspezifischen Unterschiede in der Schule zu reduzieren und der existieren Genderkluft im Informatikberufssektor entgegenzuwirken.

9. Literaturverzeichnis

- Allaire-Duquette, G., Chastenay, P., Bouffard, T., Bélanger, S. A., Hernandez, O., Mahhou, M. A., Giroux, P., McMullin, S. & Desjarlais, E. (2022). Gender Differences in Self-efficacy for Programming Narrowed After a 2-h Science Museum Workshop. *Canadian Journal of Science, Mathematics and Technology Education*, 22(1), 87–100. <https://doi.org/10.1007/s42330-022-00193-7>
- Altrichter, H. & Posch, P. (2007). *Lehrerinnen und Lehrer erforschen ihren Unterricht : Unterrichtsentwicklung und Unterrichtsevaluation durch Aktionsforschung* (4., überarb. u. erw. Aufl.). Klinkhardt. <https://ubdata.univie.ac.at/AC05772351>
- Bandura, A. (1978). Self-efficacy: Toward a unifying theory of behavioral change. *Advances in behaviour research and therapy*, 1, 161. [https://doi.org/10.1016/0146-6402\(78\)90002-4](https://doi.org/10.1016/0146-6402(78)90002-4)
- Bandura, A. (1997). *Self-efficacy : the exercise of control*. W.H. Freeman and Company. <https://ubdata.univie.ac.at/AC01935189>
- Bandura, A., Adams, N. E., Hardy, A. B. & Howells, G. N. (1980). Tests of the generality of self-efficacy theory. *Cognitive therapy and research*, 4(1), 39–66.
- Bräuer, G. (2016). *Das Portfolio als Reflexionsmedium für Lehrende und Studierende* (2., erweiterte Auflage). utb / 4141. *Schlüsselkompetenzen Hochschuldidaktik Kompetent lehren / Band VI*. Verlag Barbara Budrich. <http://www.elibrary.utb.de/doi/book/10.36198/9783838546322>
- Chinchua, S. (2022). Increasing Programming Self-Efficacy (PSE) Through a Problem-Based Gamification Digital Learning Ecosystem (DLE) Model. *Journal of higher education theory and practice*, 22(9), 131–143.
- Code with Google. *Blockly Games: About*. <https://blockly.games/about?lang=de>
- Esslinger-Hinz, I. & Sliwka, A. (2011). *Schulpädagogik. Bachelor, Master*. Beltz. <https://ubdata.univie.ac.at/AC08744739>
- Fildes, J. (14. Mai 2007). Free tool offers 'easy' coding. *BBC News*. <http://news.bbc.co.uk/2/hi/technology/6647011.stm>
- Girnat, B., Hagenauer, G. & Hascher, T. (2020). Der Zusammenhang von Schülermotivation und Mathematikleistung in der Überprüfung des Erreichens der Grundkompetenzen (ÜGK) 2016. *Schweizerischen Zeitschrift für Bildungswissenschaften*, 42, 438. <https://doi.org/10.24452/sjer.42.2.7>
- Guenaga, M. (2022). The Impact of Female Role Models Leading a Group Mentoring Program to Promote STEM Vocations among Young Girls. *Sustainability (Basel, Switzerland)*, 14(3), 1420.
- Kriegler-Kastelic, G. (2018). *Selbstwirksamkeitserwartungen: Infopool besser lehren*. Universität Wien. <https://infopool.univie.ac.at/startseite/universitaeres-lehren-lernen/selbstwirksamkeitserwartungen/>
- Lu, Y.-L. & Lien, C.-J. (2020). Are They Learning or Playing? Students' Perception Traits and Their Learning Self-Efficacy in a Game-Based Learning Environment. *Journal of educational computing research*, 57, 1909. <https://doi.org/10.1177/0735633118820684>
- Mayring, P. & Fenzl, T. (2014). Qualitative Inhaltsanalyse. In N. Baur & J. Blasius (Hrsg.), *Handbuch Methoden der empirischen Sozialforschung* (S. 543–556). Springer. https://doi.org/10.1007/978-3-531-18939-0_38

- Meluso, A. (2012). Enhancing 5th graders' science content knowledge and self-efficacy through game-based learning. *Computers and education*, 59(2), 497–504.
- Minnick, C., Abraham, N., Burd, B., Holland, E., Massaron, L. & Mueller, J. P. (2023). *Coding : alles-in-einem-Band für Dummies* (1. Auflage). ... für Dummies Lernen einfach gemacht. Wiley-VCH GmbH.
<https://ubdata.univie.ac.at/AC16863997>
- Morales-Navarro, L., Giang, M. T., Fields, D. A. & Kafai, Y. B. (2023). Connecting beliefs, mindsets, anxiety and self-efficacy in computer science learning: an instrument for capturing secondary school students' self-beliefs. *Computer science education, ahead-of-print*, 27.
<https://doi.org/10.1080/08993408.2023.2201548>
- Müller, H. & Weichert, F. (2017). *Vorkurs Informatik : der Einstieg ins Informatikstudium* (5., erweiterte und überarbeitete Auflage). Lehrbuch. Springer Vieweg. <https://ubdata.univie.ac.at/AC13770709>
- Palmer, J. D. & Rao, J. (Hrsg.) (2019, Oktober). *Pair Programming and Self-Efficacy in CS1*.
- Schön, D. A. (1983/2016). *the Reflective Practitioner: How Professionals think in action*.
<https://rauterberg.employee.id.tue.nl/lecturenotes/DDM110%20CAS/Schoen-1983%20Reflective%20Practitioner.pdf> (Erstveröffentlichung 1983)
- Schwarzer, R. & Jerusalem, M. (1995). Generalized Self-Efficacy scale. In J. Weinmann, S. Wright & M. Johnston (Hrsg.), *Measures in health psychology: A user's portfolio. Causal and control beliefs* (S. 35–37). NFER-NELSON.
- Schwarzer, R. & Jerusalem, M. (1999). *Skalen zur Erfassung von Lehrer- und Schülermerkmalen: Dokumentation der psychometrischen Verfahren im Rahmen der Wissenschaftlichen Begleitung des Modellversuchs Selbstwirksame Schulen*. Freie Universität Berlin. <http://userpage.fu-berlin.de/%7Ehealth/germscal.htm>
- Scratch Foundation. (2024). *Our Story*. Scratch Foundation.
<https://www.scratchfoundation.org/our-story>
- Stone, S. M. (2017). *observational learning*.
<https://www.britannica.com/science/observational-learning>
- Sullivan, K. (2015). CodePlus - Designing an after school computing programme for girls. In *Frontiers in Education 2015: Launching a new vision in engineering education : proceedings : October 21-24, 2015, Camino Real Hotel & Conference Center, El Paso, Texas* (S. 1–5). IEEE.
- Tan, J., Wu, L. & Ma, S. (2023). Collaborative dialogue patterns of pair programming and their impact on programming self-efficacy and coding performance. *British journal of educational technology*, 55(3), 1060–1081.
- Tellhed, U., Björklund, F. & Kallio Strand, K. (2022). Sure I can code (but do I want to?). Why boys' and girls' programming beliefs differ and the effects of mandatory programming education. *Computers in Human Behavior*, 135, 107370. <https://doi.org/10.1016/j.chb.2022.107370>
- Wei, X., Lin, L., Meng, N., Kong, S.-C. & Kinshuk (2021). The effectiveness of partial pair programming on elementary school students' Computational Thinking skills and self-efficacy. *Computers and education*, 160, 104023.

Weinhardt, F. (2017). Ursache für Frauenmangel in MINT-Berufen? Mädchen unterschätzen schon in der fünften Klasse ihre Fähigkeiten in Mathematik. *Wochenbericht - Deutsches Institut für Wirtschaftsforschung*, 84, 1014.

10. Anhang

10.1 Fragebogen zur Selbstwirksamkeit im Programmieren

Fragebogen

Einführung in die Programmierung

Name	
Klasse	
Alter	Jahre
Geschlecht	☐ männlich ☐ weiblich ☐ divers

Kreuze an!

	Gar nicht wahr	Kaum wahr	Mäßig wahr	Vollkom- men wahr
1. Ich schaffe es, schwierige Probleme beim Programmieren zu lösen, wenn ich mich ausreichend anstrengende.				
2. Wenn ich beim Programmieren auf Hindernisse stoße, kann ich Mittel und Wege finden, um mein Ziel zu erreichen.				
3. Es fällt mir leicht, an meinen Zielen beim Programmieren festzuhalten und diese zu erreichen.				
4. Ich bin zuversichtlich, dass ich mit unerwarteten Ereignissen beim Programmieren effizient umgehen kann.				
5. Dank meines Einfallsreichtums weiß ich, dass ich mit unvorhergesehenen Situationen beim Programmieren umgehen kann.				
6. Ich kann die meisten Probleme beim Programmieren lösen, wenn ich die nötige Anstrengung aufbringe.				
7. Ich kann ruhig bleiben, wenn es beim Programmieren schwierig wird, weil ich mich auf meine Bewältigungsfähigkeiten verlassen kann.				
8. Wenn ich mit einem Problem beim Programmieren konfrontiert werde, kann ich meist mehrere Lösungen finden.				
9. Wenn ein Programm nicht funktioniert, kann ich normalerweise eine Lösung finden.				
10. Ich kann mit allem umgehen, was beim Programmieren auf mich zukommt.				

10.2 Unterrichtskonzept während der Forschung

EINFÜHRUNG IN DAS PROGRAMMIEREN – Konzept

Min	Stundenplanung 1	Verbesserung 1	Verbesserung 2																
10'	Fragebogen Selbstwirksamkeit																		
	Unplugged 1																		
15'	Einstieg Strichmensch auf die Tafel zeichnen – SuS dürfen einen Namen geben Typischer Tagesablauf auf der Tafel mit SuS erarbeiten. <table><tr><td>1. Wecker läutet</td><td>9. Essen</td></tr><tr><td>2. Aufstehen</td><td>10. Hausaufgaben</td></tr><tr><td>3. Zähne putzen</td><td>11. Sport</td></tr><tr><td>4. Gesicht waschen</td><td>12. Duschen</td></tr><tr><td>5. Frühstück</td><td>13. Abendessen</td></tr><tr><td>6. Schule</td><td>14. Serien schauen</td></tr><tr><td>7. Essen</td><td>15. Zähne putzen</td></tr><tr><td>8. Schule</td><td>16. Schlafen gehen</td></tr></table> Nun darauf eingehen • Eine Person für die das gilt → Einführung Objekt • AUFGABE SuS sollen Tagesablauf für sich machen (auf Zettel) • Es wiederholt sich immer und immer wieder → Schleife • AUFGABE Was könnte man noch in einer Schleife zusammenfassen? • AUFGABE Wie oft gehört die Schleife wiederholt? ABGABE von Dokument Tagesablauf – 1x OHNE und 1x MIT Schleifen	1. Wecker läutet	9. Essen	2. Aufstehen	10. Hausaufgaben	3. Zähne putzen	11. Sport	4. Gesicht waschen	12. Duschen	5. Frühstück	13. Abendessen	6. Schule	14. Serien schauen	7. Essen	15. Zähne putzen	8. Schule	16. Schlafen gehen	Arbeitsblatt mit Fragen: Wer/was ist das Objekt bzw. für wen ist der Tagesablauf? Warum verwendet man Schleifen? → Veränderte Abgabe	Erarbeitung Schleife: Betonen, dass es wichtig ist immer genau zu definieren, was und wie oft es wiederholt werden sollte
1. Wecker läutet	9. Essen																		
2. Aufstehen	10. Hausaufgaben																		
3. Zähne putzen	11. Sport																		
4. Gesicht waschen	12. Duschen																		
5. Frühstück	13. Abendessen																		
6. Schule	14. Serien schauen																		
7. Essen	15. Zähne putzen																		
8. Schule	16. Schlafen gehen																		

	Game Based Learning 1		
15'	Spiel Lightbot Link auf Eduvidual unter „Einführung in die Programmierung“ ABGABE von Screenshot der erreichten Level <u>Extra erwähnen</u> unbedingt Schleifen verwenden (kann man an Level erkennen, min. Silber Medaille als Vorgabe geben)		
	Reflexion		
10'	Lerntagebuch - Beschreibe einen Zeitpunkt im Unterricht, bei dem du dich besonders herausgefordert gefühlt hast. Wie hast du diese Herausforderung gemeistert und welche Strategien hast du angewendet, um erfolgreich zu sein? - Reflektiere über einen Moment während des Programmierens oder Vorübungen des Programmierens, indem du ein Erfolgserlebnis hattest. Was hat dazu beigetragen, dass du dich kompetent und selbstwirksam gefühlt hast, und wie kannst du dieses Gefühl in zukünftigen Programmieraufgaben nutzen?	Veränderung der ersten Frage auf Basis des Reflexionskreislauf nach Gibbs - Beschreibe einen Zeitpunkt im Unterricht, indem du dich herausgefordert gefühlt hast. - Was hast du dabei gedacht oder gefühlt? - Was war gut und was war schlecht bezüglich der Erfahrung? - Welchen Sinn kannst du aus der Situation ziehen? - Was hättest du anders machen können? - Wie kannst du diese Erfahrung in zukünftigen Programmieraufgaben nutzen?	

Min	Stundenplanung 2	Verbesserung 1	Verbesserung 2												
15'	Wiederholung letzter Stunde Wiederholung Lightbot Zum Stundeneinstieg 10 min Lightbot weiterarbeiten lassen ABGABE von Screenshot mit entweder verbesserten Level bzw. Höheren erreichte Level BEWERTUNG – kann man dann individuelle Steigerung einfließen lassen	Schüler*innen aufmerksam machen, ob sie eine individuelle Steigerung erreicht, haben													
	Unplugged 2														
10'	Wenn Funktion Beispiel Einstieg – Zusatz Wochenende - Tagesablauf Wochenende erstellen <table><tr><td>1. Ausschlafen</td><td>6. Freunde treffen</td></tr><tr><td>2. Zähne putzen</td><td>7. Abendessen</td></tr><tr><td>3. Frühstück</td><td>8. Spieleabend</td></tr><tr><td>4. Lesen</td><td>9. Dusche</td></tr><tr><td>5. Mittagessen</td><td>10. Zähneputzen</td></tr><tr><td></td><td>11. Schlafen gehen</td></tr></table> - Unterschied Wochentage (Schule/nicht Schule) → Wenn-Funktion Variabel Beispiel Einstieg - Handelt es sich um einen Wochentag oder nicht? - Worin unterscheiden sich die zwei -> Variabel Schule	1. Ausschlafen	6. Freunde treffen	2. Zähne putzen	7. Abendessen	3. Frühstück	8. Spieleabend	4. Lesen	9. Dusche	5. Mittagessen	10. Zähneputzen		11. Schlafen gehen	Schüler*innen auffordern, ihre eigenen Beispiele für "Wenn"-Funktionen zu erstellen und zu teilen.	
1. Ausschlafen	6. Freunde treffen														
2. Zähne putzen	7. Abendessen														
3. Frühstück	8. Spieleabend														
4. Lesen	9. Dusche														
5. Mittagessen	10. Zähneputzen														
	11. Schlafen gehen														

	Game Based Learning 2		
15'	Blocky Spiele – Labyrinth Link auf Eduvidual unter „Einführung in die Programmierung“ ABGABE von Screenshot der erreichten Level (Gut, weil SuS werden gezwungen Schleifen, Wenn-Funktionen zu verwenden, und es wird Ablauf von Programm visuell dargestellt)	Integration von Peer-Feedback-Sitzungen, in denen die Schüler*innen sich gegenseitig unterstützen und konstruktive Rückmeldungen zu ihren Spielen geben können.	
Reflexion			
10'	Lerntagebuch Absatz 1 - Beschreibe einen Zeitpunkt im Unterricht, indem du dich herausgefordert gefühlt hast. - Was hast du dabei gedacht oder gefühlt? - Was war gut und was war schlecht bezüglich der Erfahrung? - Welchen Sinn kannst du aus der Situation ziehen? - Was hättest du anders machen können? - Wie kannst du diese Erfahrung in zukünftigen Programmieraufgaben nutzen? Absatz2 - Reflektiere über einen Moment während des Programmierens oder Vorübungen des Programmierens, indem du ein Erfolgs-erlebnis hattest. Was hat dazu beigetragen, dass du dich kompetent und selbstwirksam gefühlt hast, und wie kannst du dieses Gefühl in zukünftigen Programmieraufgaben nutzen?	Lerntagebuch Reflektiere über einen Moment während des Programmierens oder Vorübungen des Programmierens, indem du ein Erfolgs-erlebnis hattest. Was hat dazu beigetragen, dass du dich kompetent und selbstwirksam gefühlt hast, und wie kannst du dieses Gefühl in zukünftigen Programmieraufgaben nutzen? Falls du kein Erfolgs-erlebnis hattest, beschreibe einen Zeitpunkt im Unterricht, indem du dich herausgefordert gefühlt hast. Was war gut und was war schlecht bezüglich der Erfahrung? Welchen Sinn kannst du aus der Situation ziehen? Was hättest du anders machen können? Wie kannst du diese Erfahrung in zukünftigen Programmieraufgaben nutzen?	Individuelles Feedback zu den einzelnen Reflexion Reflektiere über einen Moment während des Programmierens oder Vorübungen des Programmierens in dieser Unterrichtseinheit, indem du ein Erfolgs-erlebnis hattest, und beschreibe diesen genau. Wann hattest du ein Erfolgs-erlebnis? Was hat dazu beigetragen, dass du dich kompetent und selbstwirksam gefühlt hast, und wie kannst du dieses Gefühl in zukünftigen Programmieraufgaben nutzen?

Min	Stundenplanung 3	Verbesserung 1	Verbesserung 2
	Einführung		
5'	Zum Start die vorprogrammierte interaktive Geschichte von Ada Lovelace herzeigen -> Wie man sowas programmiert, lernen wir heute		
	Erklärung Scratch + Pair Programming		
10'	- Scratch erklären mit Beispiel Apfel-Hunger (Objekte, Bühnenbilder, Koordinatensystem, Befehle) - Erklärung wie Pair Programming funktioniert mit Arbeitsblatt Von SuS aus dem letzten Jahr für Quelle Stellvertretene Erfahrung (2 Mädchen/1 Junge) herzeigen	Schüler*in, die schon Scratch aus dem private Umfeld kann, hilft beim Erklären	Schüler*in bedient Computer, Lehrperson erklärt am projizierten Bild
	Geschichtenentwurf		
5'	Überlegen und kurze Beschreibung einer interaktiven Geschichte in Word-Dokument Vorlage Mindestanforderung: Auswahl des Bühnenbildes, zwei Figuren, Bewegung mit Richtung und Geschwindigkeit, zwei Dialoge		
	Pair Programming		
20'	Einteilung der SuS anhand der Punkte beim Fragebogen der Selbstwirksamkeit Erstellung und Programmierung des Scratch Projektes im Pair Programming Modus: Alle 5 Minuten Timer - Wechsel Driver und Navigator	Lehrperson durchgehen und individuelles Feedback an jede Gruppe	

	Reflexion Einheit 3		
10'	Lerntagebuch Reflektiere über einen Moment während des Programmierens oder Vorübungen des Programmierens in dieser Unterrichtseinheit, indem du ein Erfolgs-erlebnis hattest, und beschreibe diesen genau. Wann hattest du ein Erfolgs-erlebnis? Was hat dazu beigetragen, dass du dich kompetent und selbstwirksam gefühlt hast, und wie kannst du dieses Gefühl in zukünftigen Programmieraufgaben nutzen? Falls du kein Erfolgs-erlebnis hattest, beschreibe einen Zeitpunkt im Unterricht, indem du dich herausgefordert gefühlt hast. Was war gut und was war schlecht bezüglich der Erfahrung? Welchen Sinn kannst du aus der Situation ziehen? Was hättest du anders machen können? Wie kannst du diese Erfahrung in zukünftigen Programmieraufgaben nutzen?		

Min	Stundenplanung 4	Verbesserung 1	Verbesserung 2
	Einführung		
5'	Mündliche Wiederholung des Stoffes der letzten Einheit		
	Pair Programming		
30'	Erstellung und Programmierung des Scratch Projektes im Pair Programming Modus: Alle 5 Minuten Timer - Wechsel Driver und Navigator Lehrperson wechselt zwischen den Teams und gibt individuelles Feedback		
	Reflexion Einheit 4		
15'	Überschrift des Lerntagebuch-Eintrages: Vorbild und Pair-Programming 1. Denke darüber nach, ob deine Lehrperson ein Vorbild für dich ist, wenn es darum geht, Programmieren zu lernen, und wie das deine Zuversicht beeinflusst, dass du es schaffen kannst. Macht es einen Unterschied für dich, ob deine Lehrperson weiblich oder männlich ist? Schreibe deine Gedanken dazu auf. 2. Beschreibe einen Moment im Unterricht, in dem du durch deinen Glauben an dich selbst etwas erreicht hast. Was lief gut und was lief weniger gut in dieser Situation? Welche Bedeutung kannst du aus dieser Erfahrung ziehen? Gibt es Dinge, die du in Zukunft anders machen würdest? Wie kannst du diese Erfahrung in zukünftigen Programmieraufgaben nutzen?		

Min	Stundenplanung 5	Verbesserung 1	Verbesserung 2
	Geschichten Präsentationen + Feedback		
20'	Wiederholung gutes Feedback- Fett markierte Wörter auf Tafel schreiben konstruktiv: Feedback sollte darauf abzielen, die Präsentationen zu verbessern, anstatt die Schülerinnen und Schüler zu kritisieren. Konstruktive Kritik sollte spezifisch, klar und hilfreich sein. Fokus auf Stärken: Beginne das Feedback mit positiven Kommentaren und Lob für gelungene Aspekte der Präsentation. Ermutige die Schülerinnen und Schüler, ihre Stärken zu erkennen und zu schätzen. Respektvoll: Achte darauf, dass das Feedback respektvoll und wohlwollend formuliert ist. Vermeide abwertende oder negative Äußerungen, die das Selbstwertgefühl der Schülerinnen und Schüler beeinträchtigen könnten. Spezifisch: Gebt konkrete Beispiele für die Aspekte, die gut waren, und für Bereiche, die verbessert werden könnten. Verwende klare und präzise Sprache, um das Feedback verständlich zu machen.	SuS als erstes aufzählen lassen Anforderung: mindestens eine Stärke nennen Gemeinsam ein gutes Feedback der Geschichte von Ada Lovelace geben	
	Feedbackrunde Die SuS öffnen ihre fertige Geschichte am PC. Die Schüler*innen teilen sich im Team in Präsentator*in und Zuseher*in auf. Die Präsentatorin bleibt am Tisch und präsentiert die Geschichte, währenddessen geht die Zuseher*in einen Tisch weiter und schaut sich eine andere Geschichte an und gibt Feedback Nach 5 min wechselt die Zuseher*in eines weiter. Bei der Hälfte ist Rollentausch	Zwei 2er Teams gehen zusammen und geben sich gegenseitig Feedback	Feedback unter Peers weglassen, dafür ihnen mehr Zeit zum Weiterarbeiten geben und dafür jede Gruppe gesondert Feedback von der Lehrperson
	Reflexion		
10'	Lerntagebuch Welcher Moment während der Präsentation deiner Scratch-Geschichte hat dir gezeigt, dass du in der Lage bist, Probleme zu lösen und deine Ideen erfolgreich umzusetzen? Wie hat dich dieser Moment in deinem Glauben an deine Fähigkeit gestärkt, als Programmierer*in Herausforderungen zu bewältigen?		
20'	FRAGEBOGEN		

10.3 Finales Unterrichtskonzept

Min	Stundenplanung 1																	
10'	Fragebogen Selbstwirksamkeit																	
	Unplugged 1																	
7'	Einstieg Die Lehrperson malt einen Strichmensch auf die Tafel. Die Schüler*innen dürfen dem Strichmensch einen Namen geben. Nun wird ein typischer Tagesablauf eine*r Schüler*in auf der Tafel gemeinsam mit den Schüler*innen erarbeitet. Beispiel: (wichtig, dass Wiederholungen enthalten sind) <table><tr><td>1. Wecker läutet</td><td>9. Essen</td></tr><tr><td>2. Aufstehen</td><td>10. Hausaufgaben</td></tr><tr><td>3. Zähne putzen</td><td>11. Sport</td></tr><tr><td>4. Gesicht waschen</td><td>12. Duschen</td></tr><tr><td>5. Frühstück</td><td>13. Abendessen</td></tr><tr><td>6. Schule</td><td>14. Serien schauen</td></tr><tr><td>7. Essen</td><td>15. Zähne putzen</td></tr><tr><td>8. Schule</td><td>16. Schlafen gehen</td></tr></table> Nun geht die Lehrperson auf folgende Punkte ein: • Eine Person für die das gilt → Einführung Objekt • Es wiederholt sich immer und immer wieder → Einführung Schleife Achtung, betonen, dass es wichtig ist immer genau zu definieren, was und wie oft es wiederholt werden sollte • FRAGE an SuS: Was könnte man noch in einer Schleife zusammenfassen? • FRAGE an SuS: Wie oft gehört die Schleife wiederholt? Erklärung des Arbeitsblattes (Word-Dokumentes) und Auftrag dieses zu lösen.		1. Wecker läutet	9. Essen	2. Aufstehen	10. Hausaufgaben	3. Zähne putzen	11. Sport	4. Gesicht waschen	12. Duschen	5. Frühstück	13. Abendessen	6. Schule	14. Serien schauen	7. Essen	15. Zähne putzen	8. Schule	16. Schlafen gehen
1. Wecker läutet	9. Essen																	
2. Aufstehen	10. Hausaufgaben																	
3. Zähne putzen	11. Sport																	
4. Gesicht waschen	12. Duschen																	
5. Frühstück	13. Abendessen																	
6. Schule	14. Serien schauen																	
7. Essen	15. Zähne putzen																	
8. Schule	16. Schlafen gehen																	
	Arbeitsblatt																	
8'	Aufgabenstellung Schreibe deinen vereinfachten Tagesablauf unter der Schulwoche hier auf (muss nicht genau der Wahrheit entsprechen). Bitte orientiere dich an dem Beispiel aus der Stunde. Beantworte anschließend die Fragen unten. Anforderungen: • min. 10 Tagesablauf-Punkte • min. 1 neue Schleife im Tagesablauf <table><tr><td>Tagesablauf ohne Schleife</td><td>Tagesablauf mit Schleife</td></tr><tr><td>1.</td><td>1.</td></tr></table> <i>Tipp: Kopieren des Tagesablaufes mit den Tasten strg+c und einfügen mit strg+v</i>		Tagesablauf ohne Schleife	Tagesablauf mit Schleife	1.	1.												
Tagesablauf ohne Schleife	Tagesablauf mit Schleife																	
1.	1.																	

	Fragen Wer/was ist das Objekt bzw. für wen ist der Tagesablauf? Warum verwendet man Schleifen? ABGABE des Arbeitsblattes auf der Lernplattform
	Game Based Learning 1
15'	Spiel Lightbot Link auf der Lernplattform - https://lightbot.lu/ ABGABE: von Screenshot der erreichten Level <u>Extra erwähnen</u> unbedingt Schleifen verwenden (Die Verwendung von Schleifen kann man an den Level erkennen, min. Silber Medaille als Vorgabe geben)
	Reflexion
10'	Lerntagebuch auf Eduvidual Reflektiere über einen Moment während des Programmierens oder Vorübungen des Programmierens in dieser Unterrichtseinheit, indem du ein Erfolgserlebnis hattest, und beschreibe diesen genau. Wann hattest du ein Erfolgserlebnis? Was hat dazu beigetragen, dass du dich kompetent und selbstwirksam gefühlt hast, und wie kannst du dieses Gefühl in zukünftigen Programmieraufgaben nutzen? Falls du kein Erfolgserlebnis hattest, beschreibe einen Zeitpunkt im Unterricht, indem du dich herausgefordert gefühlt hast. Was war gut und was war schlecht bezüglich der Erfahrung? Welchen Sinn kannst du aus der Situation ziehen? Was hättest du anders machen können? Wie kannst du diese Erfahrung in zukünftigen Programmieraufgaben nutzen?

Min	Stundenplanung 2												
	Einstieg												
15'	Wiederholung letzter Stunde Wiederholung Lightbot Zum Stundeneinstieg dürfen die Schüler*innen 10 min an Lightbot weiterarbeiten. ABGABE von Screenshot mit entweder verbesserten Leveln bzw. höher erreichte Level Betonung: Die Schüler*innen aufmerksam machen, ob sie eine individuelle Steigerung erreicht, haben Möglichkeit für die BEWERTUNG – hier kann man die individuelle Steigerung einfließen lassen												
	Unplugged 2												
10'	Wenn Funktion Beispiel vom Einstieg wird erweitert mit dem Zusatz Wochenende - Nun wird wieder gemeinsam mit den Schüler*innen der Tagesablauf einer*eines Schüler*in am Wochenende erstellt Beispiel: <table border="1"> <tbody> <tr> <td>1. Ausschlafen</td><td>6. Freunde treffen</td></tr> <tr> <td>2. Zähne putzen</td><td>7. Abendessen</td></tr> <tr> <td>3. Frühstück</td><td>8. Spieleabend</td></tr> <tr> <td>4. Lesen</td><td>9. Dusche</td></tr> <tr> <td>5. Mittagessen</td><td>10. Zähneputzen</td></tr> <tr> <td></td><td>11. Schlafen gehen</td></tr> </tbody> </table> - Nun wird auf den Unterschied der Wochentage (Schule/nicht Schule) eingegangen, → Wenn-Funktion: In der Formulierung einer Wenn-Funktion ausdrücken - Schüler*innen auffordern, ihre eigenen Beispiele für "Wenn"-Funktionen zu erstellen und zu teilen. Variabel Beispiel Einstieg - Handelt es sich um einen Wochentag oder nicht? - Worin unterscheiden sich die zwei -> Variabel Schule	1. Ausschlafen	6. Freunde treffen	2. Zähne putzen	7. Abendessen	3. Frühstück	8. Spieleabend	4. Lesen	9. Dusche	5. Mittagessen	10. Zähneputzen		11. Schlafen gehen
1. Ausschlafen	6. Freunde treffen												
2. Zähne putzen	7. Abendessen												
3. Frühstück	8. Spieleabend												
4. Lesen	9. Dusche												
5. Mittagessen	10. Zähneputzen												
	11. Schlafen gehen												
	Game Based Learning 2												
15'	Blocky Spiele – Labyrinth Link der Lernplattform https://blockly.games/maze?lang=de ABGABE: Screenshot der erreichten Level Sinn: Die Schüler*innen werden gezwungen Schleifen, Wenn-Funktionen zu verwenden, und es wird zusätzlich der Ablauf von Programm visuell dargestellt Integration von Peer-Feedback-Sitzungen, in denen die Schüler*innen sich gegenseitig unterstützen und konstruktive Rückmeldungen zu ihren Spielen geben können.												

	Reflexion
10'	Lerntagebuch auf der Lernplattform Reflektiere über einen Moment während des Programmierens oder Vorübungen des Programmierens in dieser Unterrichtseinheit, indem du ein Erfolgserlebnis hattest, und beschreibe diesen genau. Wann hattest du ein Erfolgserlebnis? Was hat dazu beigetragen, dass du dich kompetent und selbstwirksam gefühlt hast, und wie kannst du dieses Gefühl in zukünftigen Programmieraufgaben nutzen? Falls du kein Erfolgserlebnis hattest, beschreibe einen Zeitpunkt im Unterricht, indem du dich herausgefordert gefühlt hast. Was war gut und was war schlecht bezüglich der Erfahrung? Welchen Sinn kannst du aus der Situation ziehen? Was hättest du anders machen können? Wie kannst du diese Erfahrung in zukünftigen Programmieraufgaben nutzen? Lehrperson kann Individuelles Feedback zu den einzelnen Reflexion geben

Min	Stundenplanung 3
	Einführung
5'	Zum Start die vorprogrammierte interaktive Geschichte von Ada Lovelace herzeigen -> Aussage: Wie man sowas programmiert, lernen wir heute
	Erklärung Scratch + Pair Programming
10'	- Scratch erklären mit Beispiel Apfel-Hunger (Objekte, Bühnenbilder, Koordinatensystem, Befehle) - Schüler*in, die schon Scratch aus dem private Umfeld kann, hilft beim Erklären - Schüler*in bedient Computer, Lehrperson erklärt am projizierten Bild - Erklärung wie Pair Programming funktioniert mit Arbeitsblatt Interaktive Geschichten von Schüler*innen aus dem letzten Jahr für Quelle Stellvertretene Erfahrung (2 Mädchen/1 Junge) herzeigen
	Geschichtenentwurf
5'	Den Schüler*innen Zeit zum Überlegen und einer kurzen Beschreibung einer interaktiven Geschichte in Word-Dokument Vorlage geben. Mindestanforderung: Auswahl des Bühnenbildes, zwei Figuren, Bewegung mit Richtung und Geschwindigkeit, zwei Dialoge
	Pair Programming
20'	- Einteilung der SuS anhand der Punkte beim Fragebogen der Selbstwirksamkeit - Erstellung und Programmierung des Scratch Projektes im Pair Programming Modus: Alle 5 Minuten Wechsel Driver und Navigator Lehrperson: Durchgehen und individuelles Feedback an jede Gruppe geben
	Reflexion Einheit 3
10'	Lerntagebuch auf der Lernplattform Reflektiere über einen Moment während des Programmierens oder Vorübungen des Programmierens in dieser Unterrichtseinheit, indem du ein Erfolgserebnis hattest, und beschreibe diesen genau. Wann hattest du ein Erfolgserebnis? Was hat dazu beigetragen, dass du dich kompetent und selbstwirksam gefühlt hast, und wie kannst du dieses Gefühl in zukünftigen Programmieraufgaben nutzen? Falls du kein Erfolgserebnis hattest, beschreibe einen Zeitpunkt im Unterricht, indem du dich herausgefordert gefühlt hast. Was war gut und was war schlecht bezüglich der Erfahrung? Welchen Sinn kannst du aus der Situation ziehen? Was hättest du anders machen können? Wie kannst du diese Erfahrung in zukünftigen Programmieraufgaben nutzen?

Min	Stundenplanung 4
	Einführung
5'	Mündliche Wiederholung des Stoffes der letzten Einheit
	Pair Programming
30'	Erstellung und Programmierung des Scratch Projektes im Pair Programming Modus: Alle 5 Minuten Timer - Wechsel Driver und Navigator Lehrperson wechselt zwischen den Teams und gibt individuelles Feedback Abgabe auf der Lernplattform
	Reflexion Einheit 4
15'	Lerntagebuch auf der Lernplattform Reflektiere über einen Moment während des Programmierens oder Vorübungen des Programmierens in dieser Unterrichtseinheit, indem du ein Erfolgserlebnis hattest, und beschreibe diesen genau. Wann hattest du ein Erfolgserlebnis? Was hat dazu beigetragen, dass du dich kompetent und selbstwirksam gefühlt hast, und wie kannst du dieses Gefühl in zukünftigen Programmieraufgaben nutzen? Falls du kein Erfolgserlebnis hattest , beschreibe einen Zeitpunkt im Unterricht, indem du dich herausgefordert gefühlt hast. Was war gut und was war schlecht bezüglich der Erfahrung? Welchen Sinn kannst du aus der Situation ziehen? Was hättest du anders machen können? Wie kannst du diese Erfahrung in zukünftigen Programmieraufgaben nutzen?

Min	Stundenplanung 5
	Geschichten Präsentationen + Feedback
20'	Wiederholung guten Feedbacks Schülerinnen sollen Punkte guten Feedbacks nennen. Lehrperson: schreibt Punkte werden auf der Tafel mit Vorschläge CHATGPT: • konstruktiv: Feedback sollte darauf abzielen, die Präsentationen zu verbessern, anstatt die Schülerinnen und Schüler zu kritisieren. Konstruktive Kritik sollte spezifisch, klar und hilfreich sein. • Fokus auf Stärken: Beginne das Feedback mit positiven Kommentaren und Lob für gelungene Aspekte der Präsentation. Ermutige die Schülerinnen und Schüler, ihre Stärken zu erkennen und zu schätzen. • Respektvoll: Achte darauf, dass das Feedback respektvoll und wohlwollend formuliert ist. Vermeide abwertende oder negative Äußerungen, die das Selbstwertgefühl der Schülerinnen und Schüler beeinträchtigen könnten. • Spezifisch: Gebt konkrete Beispiele für die Aspekte, die gut waren, und für Bereiche, die verbessert werden könnten. Verwende klare und präzise Sprache, um das Feedback verständlich zu machen. Alle im Klassenverband geben ein „gutes“ Feedback der Geschichte von Ada Lovelace
	Feedbackrunde Die SuS öffnen ihre fertige Geschichte am PC. Die Schüler*innen teilen sich im Team in Präsentator*in und Zuseher*in auf. Die Präsentatorin bleibt am Tisch und präsentiert die Geschichte, währenddessen geht die Zuseher*in einen Tisch weiter und schaut sich eine andere Geschichte an und gibt Feedback. Anforderung ist mindestens eine Stärke zu nennen. Nach 5 min wechselt die Zuseher*in eines weiter. Bei der Hälfte der Zeit ist Rollentausch. Die Lehrperson nimmt auch Teil und gibt den Schüler*innen Feedback
	Reflexion
10'	Lerntagebuch auf Eduvidual Welcher Moment während der Präsentation deiner Scratch-Geschichte hat dir gezeigt, dass du in der Lage bist, Probleme zu lösen und deine Ideen erfolgreich umzusetzen? Wie hat dich dieser Moment in deinem Glauben an deine Fähigkeit gestärkt, als Programmierer*in Herausforderungen zu bewältigen?
20'	FRAGEBOGEN

10.4 Alle Ergebnisse der Fragebögen zur Einschätzung der Selbstwirksamkeit im Programmieren

Geschlecht	Alter	1. Fragebogen Durchschnittswert	2. Fragebogen Durchschnittswert	Differenz
weiblich	12	2,3	3,1	0,8
männlich	13	3,7	3,8	0,1
männlich	12	3,3	3,2	-0,1
weiblich	13	2,9	3,55	0,65
männlich	13	3,6	3,7	0,1
männlich	13	2,7	3,7	1
weiblich	13	2,8	3,7	0,9
weiblich	13	2,3	3,1	0,8
weiblich	13	2,9	3	0,1
männlich	13	3,5	3,6	0,1
weiblich	12	3	3,3	0,3
weiblich	12	3	3,2	0,2
weiblich	13	2,7	3,6	0,9
weiblich	13	3,2	3,7	0,5
männlich	13	3,1	3,5	0,4
weiblich	12	3,3	3,5	0,2
männlich	13	3	3,3	0,3
männlich	13	3,6	3,6	0
männlich	12	3,6	3,6	0
weiblich	13	3	3,2	0,2
männlich	13	3	3,3	0,3
männlich	12	2,35	2,75	0,4
weiblich	13	2,2	3,1	0,9
männlich	13	2,65	2,75	0,1
männlich	13	3	2,8	-0,2
männlich	12	2,9	2,9	0
männlich	12	3,2	3	-0,2
männlich	13	3,2	3,5	0,3
männlich	12	2,6	2,8	0,2
weiblich	12	2,4	2,45	0,05
männlich	13	2,8	3	0,2
männlich	12	3	2,9	-0,1
weiblich	13	2,3	3,1	0,8
weiblich	12	1,9	3,65	1,75
männlich	13	2,8	3,3	0,5

10.5 Antworten der offenen Freitextfragen

Hast du während der letzten Unterrichtsstunden eine Veränderung deines Vertrauens in deine Fähigkeiten beim Programmieren bemerkt? Wenn ja, beschreibe welche Faktoren dazu beigetragen haben.

Ich habe mich verbessert, ich denke das Scratch nicht sehr viel mit richtigen Programmieren zu tun hat, aber es dennoch viel Spaß bereitet

Nein, weil es mir schwer fällt.

Ein bisschen schon. Ich habe mich durch die Tipps meiner Lehrperson im Programmieren um einiges verbessert.

Ja ich habe volles Vertrauen in meine Fähigkeiten im Programmieren. Einer der Faktoren war, dass ich es geschafft habe Sounds in meine Story zu geben.

Eigentlich habe ich das nicht, da ich in der vorherigen nicht anwesend war. Dennoch habe ich gemerkt dass ich im Programmieren einen leichten Fortschritt bemerkt.

Ich habe gemerkt, dass es fast immer eine Lösung gibt. Man sollte nicht aufgeben.

Eigentlich nicht.

Ich habe mich sehr verbessert, da sie mir alles genau erklärt und auch mein Vertrauen ist größer geworden und bei ihr mach Unterricht auch Spaß da sie sich immer bereit erklärt uns neue Sachen erklärt. Noch ein Faktor ist das sie auch nicht so schwere Aufgaben gibt die leicht zu lösen sind.

Ja. Ich vertrau mir seit kurzem mehr, dass ich meine Ziele beim Programmieren zu erreichen kann und Probleme lösen kann. Ich vertraue mir mehr weil wir Programmieren immer einfacher vorkommt.

nei eigentlich nicht so sehr weil ich das schon gemacht habe und es war ziemlich leicht.

Leider habe ich keine wirkliche veränderung in meine programmier künsten nicht gemerkt da ich mich schon davor damit befasst habe, doch ich bin mir sicher das ich, hätte ich noch keine erfahrung damit gesammelt, sehr viel dazu gelernt da wir alle suasführlich erklärt bekahmen und wir es auch praktisch anwänden mussten.

Ich finde ich habe mich im Programmieren wirklich sehr verbessert. Ich habe bevor wir mit diesem Thema begonnen haben manchmal Probleme gehabt, die sich aber dann im Unterricht geklärt haben. Unsere lehrperson hat uns gefördert und uns mir völlig neue Funktionen von Scratch gezeigt. Mir wurde auch Kritik gegeben die ich aber gut in Erfahrung umwandeln konnte.

Ja, ich habe gemerkt das ich viel sicherer Aufgaben lösen kasnn in Hinnsicht auf programmieren und dass ich mich viel besser auskenne und logisch über Probleme nachdenken\lösen kann.

ja ich habe eine veränderung gemärkt dar ich mich jetzt sicherer fühle weil ich jetzt genau weiß was ich kann und weiß was ich noch lernen muss dandurch weiß ich welche aufgaben ich bewältigen kann und welche nicht.

Ja ich habe eine Veränderung Meiner fähigkeiten bemerkt da ich sehr viel sicherer im Programieren wurde und das weil ich durch übung besser wurde

Ja ich habe eine Veränderung meiner Fähigkeiten gemerkt da ich selbsbewusster wurde und meine Lehrperson nie böse war wenn wir etwas nicht gewusst haben. Meine Lehrperson hat mir immer geholfen und somit wurde ich immer besser im programmieren auf "Scratch".

Ja ich habe mehr vertrauen in meine fähigkeiten bekommen da ich es heute präsentiert habe und das feedback gut war

Nicht wirklich. Aber ich habe ein paar tips bekommen wie ich die Geschichte noch Spannender machen kann. Und wie ich die Unterhaltung besser Schreiben kann.

Ja, durch das positive Feedback meiner Mitschüler. Der Rest hat sich von letzter auf diese stunde nicht verändert.

Leider war ich in einer Stunde krank und konnte so weniger programmieren als die anderen. Sonst habe ich schon neue Sachen beim Programmieren gelernt. Jetzt habe ich schon mehr Vertrauen in meine Fähigkeit nicht sehr viel aber schon etwas.

Leider fehlte ich eine Stunde und konnte deshalb nicht so viel programmieren wie andere. Trotzdem wurde mein Vertrauen in die Fähigkeit des Programmierens sehr gestärkt.

Ich habe bei dem Programmieren vortschritte gemerkt und habe gut gelernt mit dem Programm "Scratch" umzugehen. Es hat dazu beigetragen das meine Lehrperson alles gut erklärt hat und uns bevor wir damit gearbeitet haben gut darauf vorbereitet hat.

ja auf jeden fall! ich kannte Scratch davor zwar schon, aber dieses Mal haben wir uns viel mit den Verschiedenen Möglichkeiten beschäftigt, und es geschafft eine komplexe Geschichte daraus zu entwickeln. Dazu, das wir das geschafft haben, haben sicher einerseits unser gutes Team-Arbeiten beigetragen und die freiheit die uns gegeben wurde. Wir durften einfach irgentwas machen. auf jeden Fall hat mit das programmieren gefallen.

Ich habe viel mehr Selbstvertrauen beim programmieren bekommen. Ich bin natürlich besser geworden über die letzten Stunden das hat mein Selbstvertrauen sicher verbessert. Das ich von anderen gelobt wurde hat auch dazu beigetragen.

Ich habe schon einen Unterschied bemerkt denn am anfang war bei Scratch mein Ziel das ich eine Katze zum gehen bringe, heute ist es eine ganze Geschichte.

Es fällt mir mittlerweile etwas leichter zu programmieren. Das Spiel zu erstellen hat mir ebenfalls sehr viel Spaß bereitet und hat sicherlich bei den Fortschritten im Programmieren beigetragen.

Ich habe einen Veränderung durch aus gehabt da ich viel besser in den aufgaben wurde und somit sicherer war. Faktoren: genügend Zeit, nette Lehrer, schöne Aufgaben, gut beschrieben. alles dinge die mich zum arbeiten bringen. Ich finde den Unterricht sehr interessant wenn wir immer coole dinge machen die man im Team und alleine macht.

Ich weiß jetzt besser wie programmieren geht. Ich kenne mich jetzt gut aus. Ich habe davor noch nicht so oft programmiert. Ich bin jetzt zuversichtlich das ich es schaffen zu programmieren.

nein, eigentlich habe ich meine veränderung meines vertrauens nicht geändert. es ist gleich geblieben. ich probiere es einfach und mache es.

Ja habe ich ich bin aber eher selber darauf gekommen oder mithilfe meines freunden.

Nein.

ich habe meine Fähigkeiten beim programmieren ein wenig überschätzt nur weil ich mal in einem kurs dafür war, obwohl das schon lange her ist. trotzdem habe ich gemerkt das ich kein schlechter programmierer bin. ich konnte einiges bei diesem projekt schon gut nur wie man eine figur etwas sprechen lassen kann habe ich nicht mehr gewusst.

Ja schon, weil ich vieles neues gelernt und verstanden habe. Auch habe ich den kompletten Abschnitt "Fühlen" erst so richtig verstanden.

Inwiefern spielt für dich die Lehrperson eine Rolle bei der Begleitung beim Programmieren-Lernen?

Nicht wirklich, ich kenne die basics zum programmieren daher nützt es mir persönlich nix.

Es hilft mir die Aufgabe leichter zu verstehen

Die Lehrperson ist sehr wichtig, da sie bereit sein muss den Schülern zu helfen und ihnen Sachen zu erklären. Ob männlich oder weiblich ist mir jedoch egal. Die Lehrperson muss auch bereit sein Fragen zu den Basics der Informatik Fragen zu beantworten.

Sie spielt eine große Rolle beim Programmieren-Lernen , da man sie alles fragen kann, wenn man Hilfe braucht.

Ich glaube das ich ohne die Hilfe der Frau Prof. nicht im Stande wäre einigermaßen komplexe Sachen zu programmieren, wie zum Beispiel Scratch

Bis jetzt nicht soo viel bei den einzelnen Arbeitsschritten aber beim Grundwissen schon. Sie hat uns gezeigt wie programmieren geht und wie wir nach Lösungen finden kann. Frau Professorin Schaußberger kann ich immer etwas Fragen, wenn ich etwas nicht verstehe. Sie ist für mich Vorbild, weil sie sehr gut programmieren kann, und weil sie Zeit, dass in der heutigen Gesellschaft auch „eine Frau“ mit Computern umgehen kann.

Ohne der Hilfe der Frau Prof. würde ich schwierige Situation beim Programmieren vermutlich nicht schaffen.

Für mich eine große Rolle, denn meistens wenn ich nach langem versuchen nicht weiter komme hilft sie mir meistens und verliert nicht die Geduld.

Sie hilft mir Probleme zu Lösen und erklärt mir Sachen die ich noch nicht weiß

Eigentlich mach sei mir viel aus weil wenn ich jetzt was nicht verstehe kann sie mir immer helfen

Die Lehrperson soll den Kindern bei schwierigen aufgaben helfen und ihnen dinge erklären, damit sie gut weiterarbeiten können und nicht sich zu lange damit befassen. Zum glück ist das bei uns der fall.

Als wir mit dem Programmieren begannen habe ich mich nur halbwegs ausgekannt. Ich finde die Lehrperson hat eine große Rolle gespielt da uns neue Sachen beigebracht wurden und Probleme gestellt wurden. das finde ich sehr sehr wichtig.

Es ist gut wichtig das die Lehrperson alle Fragen und Probleme beantwortet und dir hilft wenn du Schwierigkeiten hast.

Ich finde die Lehrperson muss hilfsbereit sein und die aufgaben angemessen wählen damit sie nicht zu anspruchsvoll für das derzeitige können sind . Das ist beim programmieren lernen sehr wichtig denn sonst können aufgaben schnell sehr frustrierend wirken.

Sie ist eine Ansprechperson sobald ich eine frage habe.Und mir jederzeit helfen will meine probleme zu lösen.

Wenn ich mich nicht auskenne kann ich meine Lehrperson immer fragen und sie weiß immer eine Lösung. Außerdem kann sie sehr gut erklären sodass man meist keine Fragen während dem programmieren selbst mehr hat.

Sie erklärt mir alles ohne sie würde ich gar nix verstehen. Sie ist sehr wichtig ich weiß nicht ob ich diese Aufgaben ohne sie bewältigen.

Naja sie ist halt meine Lehrerin ,sie bringt mir die Sachen bei und hilft mir wenn ich Probleme habe.

Sie spielt eine sehr große, wie eine wichtige Rolle für mich, da ich mit meinen Programierfehler immer zu der Lehrperson gehen kann um, um rat zu fragen. Ebenso kann sie mich außbessern und weitere dinge verbessern. Somit kann ein schüler immer besser und besser werden. Dadurch habe ich zuvericht, mal gut programieren zu können

Beim Programmieren haben wir Partnerarbeit gemacht, so konnten wir viele Probleme in der Gruppe lösen. Also musste die Prof. uns nicht so oft helfen. Ich finde in anderen Themen brauche ich die Lehrperson bei Fragen eher als wenn die meisten Fragen schon in der gruppe geklärt werden können.

wir haben viele Themen vom Programmieren durch genommen und uns eine längerre Zeit damit beschäftigt.Ohne unsere Lehrperson währe ich wahrscheinlich nicht so instande gewesen das Programmieren zu erlernen .Allerdings habe ich auch schon öfters mit scratch programmiert.Trotzdem hat mir meine Lehrperson immer geholfen wenn ich etwas nicht verstanden habe oder nicht in der Lage war es selbst zu lösen.

Unsere Lehrperson war für mich beim Programmieren sehr wichtig da ich ohne eine Erklärung nicht alles so gut verstanden hätte. Unsere Lehrperson hat meine Fragen auch immer gut beantwortet so das ich es dann verstanden hab.

Ich finde die Lehrperson spielt bei dem Thema schon eine große Rolle, da der Schüler ja wenig oder gar nichts vom programmieren weis. Da kann eine Lehrkraft wenn sie den Unterricht gut gestaltet schon das Interesse am Programmieren erwecken, auch wenn es davor nicht so da war. Ich finde auch, dass Schüler sich viel leichter bei einem Problem tun, wenn die Professorin sich bei der Sache gut auskennt. Auf jeden Fall hängen meiner Meinung nach die Leistungen beim programmieren viel mit der Lehrkraft zusammen.

Sie kann mir halt bei Problemen helfen und bringt uns halt so gut wie alles bei. Also spielt die Lehrperson eine große Rolle.

Ich finde eine Lehrperson spielt beim programmieren schon eine sehr große rolle, denn wie sollte ich ohne eine Lehrperson wissen was ich machen soll beim programmieren?

Eine Lehrperson beim programmieren lernen immer fragen zu können ist sehr essenziel. Da falls ich fragen habe oder Schwierigkeiten immer um Hilfe bitten kann. Auserdem festigt es das Verständniss für den/die Lernende:n schneller wenn er/sie Feedback bei Aufgaben erhalten kann.

Sie spielt durch aus eine Rolle. Den ohne eine Lehrperson wären Fragen und Probleme UM so schwerer. Ich finde bei einer guten Lehrperson lernt man gut.

Sie kann mir helfen wenn ich bei etwas nicht weiß. Und sie kann mir auch Verbesserungsvorschläge geben. Und es bewerten. Sie untertschützt mich wenn ich mal nicht weiter weiß.

die Lehrperson hilft mir sehr und ich mache das was sie mir erklärt. wenn ich Omar oder Stefan frage wie das funktioniert, erklären sie mir wie das geht aber ich verstehe es weniger als wenn es mir die Lehrperson erklärt.

Nicht so groß eher mittelmäßig.

Ich finde sie sollte die Grundlagen vorzeigen und wenn wir dann etwas eigenes Programmieren sollten soll sie uns bei den schwereren Sachen helfen. Wir sollten nach den Grundlagen selber Probieren können und selber neue Sachen entdecken können. So macht es auch mehr Spaß.

eigentlich keine sehr große da ich die kniffligen aufgaben beim programieren selber versuche zu lösen. komme ich garnicht darauf wie ich ein Problem bewältigen kann dann frage ich meistens zuerst noch andeere SchülerInnen um Rat und dan erst die Lehrerin.

Schon eine sehr Wichtige, da man noch immer soo viel nicht versteht und die Lehrperson einem immer weiterhelfen kann.

10.6 Elternbrief

Sehr geehrte Erziehungsberechtigte,

Ich werde eine Forschung für meine Masterarbeit "Erforschung eines inklusiven Unterrichtskonzeptes zur Erhöhung der Selbstwirksamkeit im Programmieren von Schüler*innen der 7. Schulstufe, insbesondere von Mädchen" im Rahmen des Informatikunterrichtes durchführen.

Für diese Studie möchte ich in der 7. Schulstufe im Regelunterricht eine Erhebung durchführen, an der Ihr Kind freiwillig teilnehmen kann. Sie würden mir sehr mit ihrer Einverständniserklärung unterstützen.

Das Ziel meiner Arbeit ist es, herauszufinden, ob geschlechtsspezifische Unterschiede in der Selbstwirksamkeit bezüglich des Programmierens existieren und ob ein spezifisches inklusives Unterrichtskonzept diese steigern kann. Die Studie soll zur Verringerung der Genderkluft im Informatiksektor beitragen und praktische Impulse für den Informatikunterricht bieten.

Die Datenerhebung erfolgt durch Fragebögen zu Beginn und Ende des Themas im Regelunterricht, ist freiwillig und kann jederzeit ohne Nachteile abgebrochen werden.

Die Daten werden anonymisiert ausgewertet und dienen ausschließlich wissenschaftlichen Zwecken. Die Schulleitung hat der Studie bereits zugestimmt.

Ich freue mich über Ihre Unterstützung. Bei Fragen stehe ich gerne zur Verfügung.

Mit freundlichen Grüßen,

Theresa Schausberger, BEd

Kontakt: theresa.schausberger@bildung.gv.at

Bitte diesen Abschnitt abtrennen und den*der Schüler*in wieder mitgeben.

Einverständniserklärung zur Teilnahme an der Forschung

Hiermit erkläre ich, _____ [Vorname Nachname des Erziehungsberechtigten], als Erziehungsberechtigte(r) von _____ [Name des Kindes], dass ich mein Einverständnis zur Teilnahme meines Kindes an der oben beschriebenen Studie gebe.

☐ JA, ich erkläre mich mit der Teilnahme meines Kindes einverstanden.

☐ NEIN, ich erkläre mich nicht mit der Teilnahme meines Kindes einverstanden.

Ich habe den Projektbeschrieb gelesen und verstanden sowie die Freiwilligkeit der Teilnahme und die Möglichkeit des jederzeitigen Widerrufs der Einwilligung zur Kenntnis genommen.

Datum: _____ Unterschrift Erziehungsberechtigte*r: _____