

Westfälische Wilhelms-Universität Münster
Institut für Didaktik der Mathematik und Informatik
Wintersemester 2021/2022

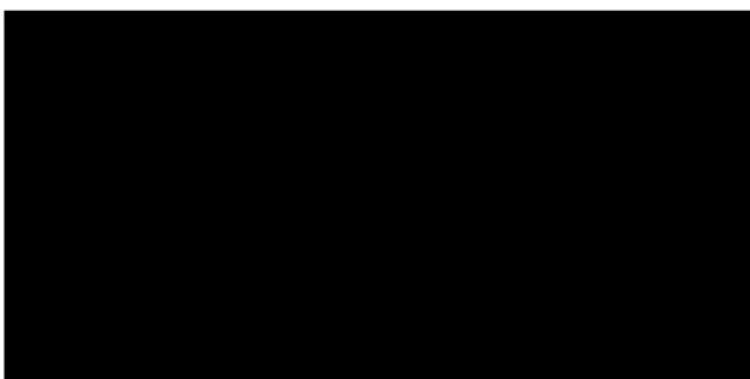
Bachelorarbeit

**Herausforderungen in der Grundschule beim Umstieg von
ScratchJr zu Scratch**

Challenges in elementary school switching from ScratchJr to Scratch

Erstgutachter: Prof. Dr. Marco Thomas
Zweitgutachter: Prof. Dr. Daniel Frischemeier

Vorgelegt von:
Marius Heinrich Maria Klasen



Inhaltsverzeichnis

1.) Einleitung	1
2.) Das Programm Scratch	2
2.1) Allgemeines	2
2.2) Aufbau	5
2.3) Handhabung.....	7
2.4) ScratchJr	9
2.4.1) Allgemeines.....	9
2.4.2) Aufbau.....	10
2.4.3) Handhabung	11
2.5) Scratch und ScratchJr im Vergleich	12
3.) Kompetenzerwerb mit Hilfe von Scratch	12
3.1) Erwerb informatischer Kompetenzen.....	13
3.2) Erwerb mathematischer Kompetenzen.....	14
4.) Umstieg von ScratchJr zu Scratch	17
4.1) Rahmenbedingungen	17
4.2) Methodik.....	18
4.2.1) Erste Hypothese	19
4.2.2) Zweite Hypothese	20
4.2.3) Dritte Hypothese	25
4.3) Ergebnisse.....	27
4.4) Analyse	30
5.) Fazit	34

6.) Literaturverzeichnis	36
7.) Abbildungsverzeichnis.....	39
8.) Anhang	40

1.) Einleitung

Informatische Kompetenzen nehmen in unserer Gesellschaft eine zunehmend wichtige Rolle ein und Informatiksysteme haben längst Einzug in die kindliche Welt gefunden. „Eine bewusste Teilnahme am Leben in unserer Gesellschaft, aber auch die konstruktive Mitgestaltung der Lebenswelt, setzen zunehmend informatische Kompetenzen voraus.“¹ Deshalb sollte der informatischen Bildung bereits in der Grundschule Raum zur Entfaltung geboten werden, damit Schüler² Probleme, die im informatischen Kontext auftreten, eigenständig lösen können.³ Einen Teil der informatischen Bildung machen dabei Algorithmen aus, welche zum Entwickeln informatischer aber auch außer-informatischer Kompetenzen beitragen können (siehe Kapitel 3.1). Der Begriff der Algorithmen hat unter anderem Einzug in den Medienkompetenzrahmen des Landes Nordrhein-Westfalen (NRW) gefunden, dessen Ziel die Umsetzung der bundesweiten Bildungsstandards im Bereich digitaler Medien ist.⁴ Darüber hinaus werden Algorithmen ebenfalls im Lehrplan Mathematik der Primarstufe des Landes NRW aufgegriffen, wodurch die Relevanz außerhalb der Informatik verdeutlicht wird.

Algorithmen können als Handlungsvorschriften oder Ablaufbeschreibungen bezeichnet werden, welche für die korrekte Ausführung sprachlich äußerst präzise formuliert sein müssen und somit auch sprachliche Kompetenzen fördern können.⁵ Algorithmen können durch verschiedene Arten von Sprache realisiert werden, wobei diese häufig mit Programmiersprachen assoziiert werden. Programmiersprachen werden häufig mit sehr komplizierten Softwareprogrammen in Verbindung gesetzt, welche eine hohe technische Affinität und informatischen Wissen des Anwenders voraussetzen. Dabei gibt es Programmiersprachen, die jüngere Zielgruppen ins Auge fassen und mit wenigen Vorkenntnissen erlernt werden können. Ein Beispiel für eine solche Programmiersprache stellt das Programm Scratch dar. Scratch ist eine explizit für Kinder und Jugendliche konzipierte Programmiersprache, die jungen Menschen das Programmieren von Geschichten, Spielen und Animationen ermöglichen soll.⁶ Neben dem Standardprogramm wurde

¹ S. Gesellschaft für Informatik e.V., 2019, S. 5.

² Aus Gründen der besseren Lesbarkeit wird in dieser Arbeit die Sprachform des generischen Maskulinums verwendet. Diese umfasst gleichermaßen die weibliche Form.

³ Vgl. Gesellschaft für Informatik e.V., 2019, S. 5.

⁴ Vgl. Medienberatung NRW et al., 2021, online unter:

<https://medienkompetenzrahmen.nrw/medienkompetenzrahmen-nrw/> (Stand 20.12.21).

⁵ Vgl. Gesellschaft für Informatik e.V., 2019, S. 13.

⁶ Vgl. Scratch Foundation, 2021, online unter: <https://scratch.mit.edu/about> (Stand 20.12.21).

zudem eine vereinfachte Version namens ScratchJr entwickelt, welche auf Kinder im Alter von fünf bis sieben Jahren zugeschnitten ist.

Studien zur Verwendung der Programme deuten auf positive Effekte im Hinblick auf informatische und mathematische Kompetenzbereiche in der (Grund-)Schule hin, weshalb Scratch bzw. ScratchJr einen Bildungsbeitrag – insbesondere im Bereich der Algorithmen – leisten können. Die bisherigen Studien fokussieren sich auf die Arbeit mit Scratch oder ScratchJr, jedoch nicht auf mögliche Auswirkungen der Verwendung beider Programmvarianten.

Ziel dieser Arbeit ist es, einen Beitrag dazu zu leisten, Herausforderungen in der Grundschule beim Umstieg von Scratch zu ScratchJr aufzuzeigen. Dafür werden drei Hypothesen durch verschiedene Arbeitsaufträge und Fragebögen innerhalb einer Arbeitsgemeinschaft (AG) an einer Grundschule überprüft. Die Hypothesen fokussieren sich dabei auf grundsätzliche Programmunterschiede, Potenziale bei der Behandlung mathematischer Inhalte sowie die Eignung der vereinfachten ScratchJr Variante für die Primarstufe.

Der Einstieg dieser Arbeit umfasst die Beschreibung von Scratch sowie ScratchJr, welche Grundsätzliches, den Aufbau, Handhabung sowie eine Gegenüberstellung der Programme umfasst. Anschließend werden die Potenziale der beiden Programme im Hinblick auf den Erwerb relevanter Kompetenzen aufgezeigt. Diese umfassen informatische Kompetenzen basierend auf den Empfehlungen der Gesellschaft für Informatik e.V. (GI) und des Medienkompetenzrahmens NRW sowie mathematische Kompetenzen auf Grundlage verschiedener Studienerkenntnisse.

Daran anknüpfend erfolgt die Überprüfung der Hypothesen. Aus den Erkenntnissen der Ergebnisanalyse sollen abschließend mögliche Herausforderungen erarbeitet werden, welche beim Übergangsprozess der Programme entstehen können.

2.) Das Programm Scratch

2.1) Allgemeines

Das Projekt Scratch wurde im Jahr 2002 durch die Lifelong Kindergarten-Gruppe, angeführt von Professor Mitchel Resnick, am MIT Media Lab initiiert und im Jahr 2007 erstmals veröffentlicht.⁷ Seit der Veröffentlichung wurde das Programm stetig entwickelt, wird seit 2019

⁷ Vgl. Scratch Foundation, 2021, online unter: <https://scratch.mit.edu/credits> (Stand 10.11.21).

von der Scratch Foundation geführt und läuft seitdem unter der aktuellsten Version 3.0.⁸ Da es sich bei der Scratch Foundation um eine Non-Profit-Organisation handelt steht die Software kostenlos zur Verfügung und läuft in den meisten Webbrowsern auf den gängigsten Endgeräten.⁹ Scratch ist eine Programmiersprache, mit welche interaktive Projekte wie Geschichten, Spiele und Animationen erstellt werden können und wodurch junge Menschen Kreativität, logisches Denken und Teamarbeit erlernen würden.¹⁰ Die Vision der Scratch Foundation ist die Verbreitung kreativer, fürsorglicher, kooperativer und gerechter Ansätze für Codierung und Lernen in der ganzen Welt, wobei die Firma ihren Auftrag darin beschreibt, junge Menschen mit digitalen Werkzeugen und Möglichkeiten auszustatten, um etwas zu schaffen, zu teilen und zu lernen.¹¹ Präziser beschreiben die Entwickler die Hintergründe von Scratch wie folgt:

We've developed Scratch as a free, safe, playful learning environment that engages all children in thinking creatively, reasoning systematically, and working collaboratively - essential skills for everyone in today's society. We work with educators and families to support children in exploring, sharing, and learning.¹²

Als Leitbild nennt die Scratch Foundation die vier P's kreativen Lernens, welche *projects*, *passion*, *peers* und *play* umfassen (Abbildung 1).

Vier P's kreativen Lernens	
Projects	Passion
Kinder zum Entwickeln, Gestalten und zum kreativen Ausdruck anleiten.	Kindern das Ausleben ihrer Interessen und die Arbeit an für sie wichtigen Projekten ermöglichen.
Peers	Play
Kinder beim Zusammenarbeiten, Teilen, Remixen und Helfen unterstützen.	Kindern zum Basteln, Experimentieren und Wiederholen ermuntern.

Abbildung 1: Vier P's kreativen Lernens¹³

Grundsätzlich kann jeder mit Scratch Projekte erstellen, wobei insbesondere junge Menschen im Alter von 8 bis 16 Jahren die Zielgruppe sind.¹⁴ Dies lässt sich auch aus den statistischen Daten

⁸ Vgl. Scratch Foundation, 2021, online unter: <https://scratch.mit.edu/faq#scratch3> (Stand 10.11.21).

⁹ Vgl. ebd. (Stand 10.11.21).

¹⁰ Vgl. ebd. (Stand 10.11.21).

¹¹ Vgl. Scratch Foundation, 2021, online unter: <https://scratch.mit.edu/annual-report> (Stand 16.12.21).

¹² S. ebd. (Stand 16.11.21).

¹³ Grafik selbst erstellt. Inhalt S. Scratch Foundation, 2021, online unter: <https://scratch.mit.edu/annual-report> (Stand 10.11.21).

¹⁴ Vgl. Scratch Foundation, 2021, online unter: <https://scratch.mit.edu/faq#scratch3> (Stand 10.11.21).

des Entwicklerteams ableiten, nach denen die meisten neuen Nutzer zwischen 10 und 14 Jahre alt sind (Abbildung 2).

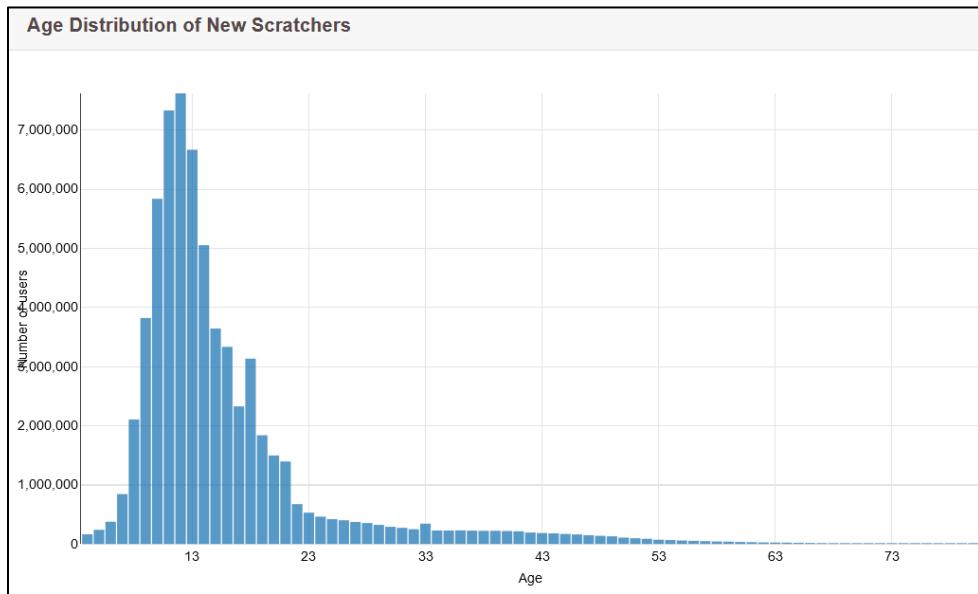


Abbildung 2: Altersverteilung neuer Scratch-Nutzer¹⁵

Scratch wurde zudem in mehr als 60 Sprachen übersetzt und wird international genutzt, wobei etwas mehr als 80 Millionen Nutzer weltweit registriert sind.¹⁶ Die Nutzungsaktivitäten von Scratch erhöhen sich laut den firmeneigenen Statistiken stetig, wobei ein stärkerer Anstieg seit März 2020 – insbesondere bei der Kommentaraktivität – zu erkennen ist.¹⁷ Hier liegt ein Zusammenhang mit dem Beginn der Corona-Pandemie nahe.

Darüber hinaus zielt Scratch besonders auf die Institution Schule als Nutzungsraum ab und bietet für Lehrkräfte und Schüler explizite Informationen und Ressourcen zur Verwendung des Programms an.¹⁸ Diese Ressourcen werden einerseits vom hauseigenen Scratch-Team und andererseits durch unterschiedliche Organisationen, Unternehmen oder Lehrkräfte auf der gesamten Welt bereitgestellt. Schülern werden dabei beispielsweise Scratch-Tutorials, Scratch-Karten und Ideen angeboten, mit denen der Einstieg und Umgang mit Scratch anhand von Anleitungen erleichtert wird. Beispielsweise bietet die Scratch Foundation einen *Educator Guide*

¹⁵ Grafik aus: Scratch Foundation, 2021, online unter: <https://scratch.mit.edu/statistics/> (Stand 10.11.21).

¹⁶ Vgl. Scratch Foundation, 2021, online unter: <https://scratch.mit.edu/statistics/> (Stand 10.11.21).

¹⁷ Vgl. ebd. (Stand 10.11.21).

¹⁸ Vgl. Scratch Foundation, 2021, online unter: <https://scratch.mit.edu/educators/> (Stand 10.11.21).

an, welcher einstündige Workshops zur Verwendung des Programms anbietet.¹⁹ Ressourcen anderer Firmen umfassen z.B. einen von Google angebotenen kostenlosen Lehrplan für das Fach Informatik, welcher die Arbeit mit Scratch umfasst.²⁰

Einen weiteren Beitrag zur Schularbeit liefern zudem die Webseiten *Scratch in Practice* und *ScratchEd*, bei denen das Lernen mit dem Programm ebenfalls im Fokus steht.²¹ *ScratchEd* spielt bei der Verbindung des Programms mit curricularen Inhalten eine wichtige Rolle, da konkrete Vorschläge für den Einsatz von Scratch in diversen Unterrichtsfächern gegeben werden.²² Scratch könnte in jeden Inhaltsbereich aller Klassenstufen aufgenommen werden und könnte in Fächern wie Englisch, Naturwissenschaften, Mathe, Geschichte oder Kunst zum Lernen der Schüler auf vielen Wegen beitragen.²³ Grundsätzlich lebt Scratch von der Vernetzung mit anderen Mitgliedern und der Weiterverwendung, Verarbeitung und Bereitstellung von Projekten anderer Nutzer der Online-Community.

Das Programm verfügt zudem über ein eigenes Wiki, welches es in verschiedenen Sprachen gibt und von Nutzern des Programms verwaltet wird.²⁴ Über das Wiki können beispielsweise Informationen über Blockkategorien oder Blockfunktionen eingeholt werden, welche einen wesentlichen Bestandteil des Programms ausmachen.

2.2) Aufbau

Scratch ist ein umfassendes Browserprogramm mit vielen Funktionen, welche zudem durch Erweiterungen anderer Firmen ausgeweitet werden können. Im Rahmen dieser Arbeit werden diese Erweiterungen nicht näher erläutert, sodass sich auf die wichtigsten Elemente des Programms fokussiert werden kann.

Sobald man in den Bearbeitungsbereich von Scratch gelangt, kann dieser im Groben in drei Bereiche unterteilt werden, welche sich links, mittig und rechts befinden (Abbildung 3). Im linken Bereich befinden sich die Registerkarten Skripte, Kostüme und Klänge. Bei Auswahl von *Skripte* können Blöcke ausgewählt werden, welche zum Erstellen von Programmierungen genutzt

¹⁹ Vgl. Scratch Foundation, 2021, online unter:

<https://resources.scratch.mit.edu/www/guides/en/EducatorGuidesAll.pdf> (Stand 10.11.21).

²⁰ Vgl. Google, 2021, online unter: <https://csfirst.withgoogle.com/s/de/home> (Stand 31.11.21).

²¹ Vgl. Scratch Foundation, 2021, online unter: <https://sip.scratch.mit.edu/themes/curriculum/> (Stand 31.11.21).

²² Vgl. Harvard Graduate School of Education, 2021, online unter: <https://scratched.gse.harvard.edu/index.html> (Stand 01.12.21).

²³ Vgl. Harvard Graduate School of Education, 2017, online unter:

<https://scratched.gse.harvard.edu/resources/scratch-across-every-subject-recap.html> (Stand 01.12.21).

²⁴ Online unter: <https://de.scratch-wiki.info/>.

werden. Beim Auswählen der Registerkarten Kostüme und Klänge können Figuren sowie Hintergründe bzw. Sounds ausgewählt werden.

Der mittlere Bereich fungiert als Bearbeitungsbereich für die Skripte als auch für die Kostüme und Klänge. Bei Auswahl der entsprechenden Registerkarte können Programmierungen, Figuren und Hintergründe sowie Klänge in diesem Bereich bearbeitet werden.

Im rechten oberen Bereich können die ausgewählten und ausgeführten Blöcke beobachtet werden, während sich im darunterliegenden Areal die ausgewählten Figuren und Bühnenbilder befinden.

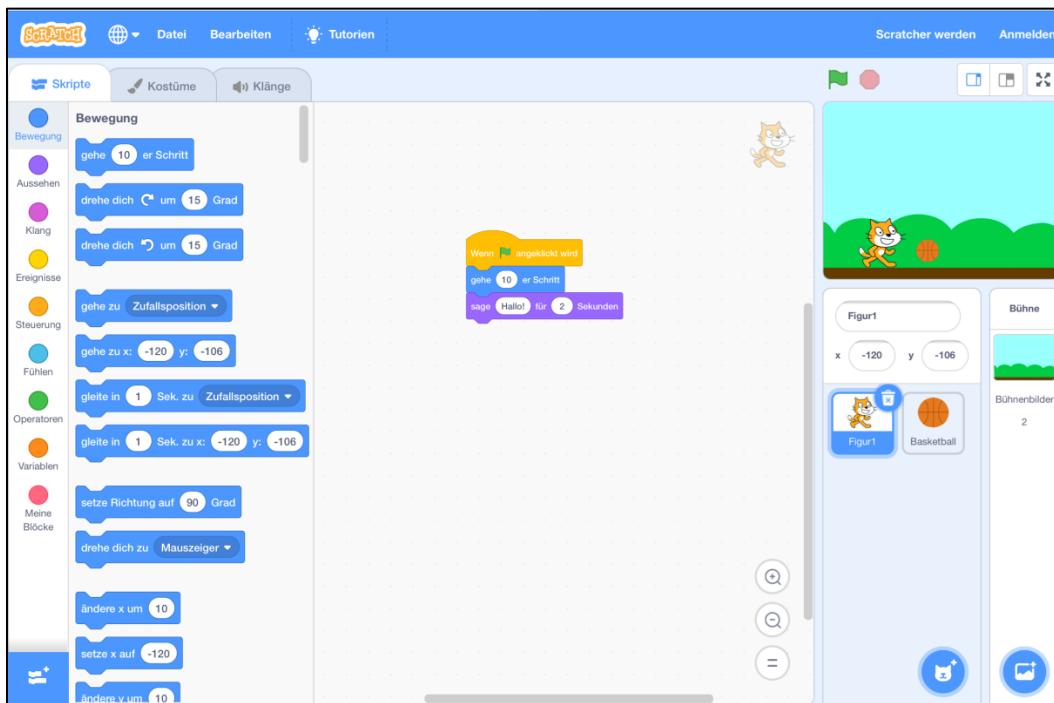


Abbildung 3: Screenshot des Browserprogramms Scratch mit einer Beispielprogrammierung.

In der Leiste über diesen Bereichen befinden sich zudem weitere Möglichkeiten, wie die Sprache zu ändern, Dateien zu speichern und hochzuladen sowie eine Auswahl von Tutorien. Letztere umfassen 26 Videoanleitungen zu ausgewählten Programmierungsideen wie die Animation des eigenen Namens oder das Entwickeln verschiedener Spiele.

Den Kern von Scratch bilden die insgesamt 107 Programmierblöcke. Die unterschiedlichen Blöcke sind nach Funktionsarten und in bestimmte Farben unterteilt, welche auf der linken Seite von Abbildung 3 zu erkennen sind.²⁵

²⁵ Eine Blockübersicht befindet sich im Anhang 4.

2.3) Handhabung

Skripte können in Scratch erstellt werden, indem die Blöcke in den mittleren Bearbeitungsbereich gezogen werden, in welchem diese durch Berührung aktiviert werden können.

Mit den blauen Blöcken führen die Figuren Bewegungen aus. Neben einfachen Gehbewegungen sind ebenfalls Drehbewegungen oder Bewegungen in Abhängigkeit von x und y Variablen durchführbar.

Die Blöcke der Kategorie Aussehen verändern das Erscheinungsbild auf unterschiedliche Weise. Beispielsweise können Sprechblasen zu Figuren hinzugefügt, Kostüme oder Bühnenbilder gewechselt oder die Größe von Objekten verändert werden.

Im violetten Bereich der Klänge wird Hörbares verwaltet, indem z.B. selbstaufgezeichnete Klänge hinzugefügt werden können, welche zudem im Grad der Lautstärke angepasst werden können.

Ereignisblöcke umfassen größtenteils „Wenn“-Blöcke, welche zu Beginn eines Algorithmus stehen, wodurch Anweisungen von bestimmten Bedingungen abhängig gemacht werden können. Beispielsweise wird eine Anweisung nur dann ausgeführt, wenn eine bestimmte Taste betätigt wird.

Bedingungen, welche nicht den Start von Blöcken bestimmen, sondern innerhalb verschiedener Blocksequenzen eingesetzt werden können, finden sich unter der Kategorie Steuerung. Innerhalb dieser Kategorie können exemplarisch Blöcke wiederholt werden oder nach einem Block für eine bestimmte Zeit gewartet werden.

Die hellblauen Fühlen-Blöcke erfragen bestimmte Zustände. In diesem Bereich können Bedingungen genauer definiert werden, indem z.B. gefragt wird, ob eine bestimmte Figur berührt oder eine bestimmte Taste gedrückt wurde.

Die grünen Blöcke umfassen vor Allem mathematische Operatoren, mit denen Zahlen und Rechengesetze implementiert werden können. Beispielsweise können die vier Grundrechenarten eingefügt werden oder Zahlen zufällig generiert werden.

Innerhalb der Kategorie der Variablen ist es möglich Größen im Algorithmus zu definieren, indem z.B. bestimmte Zahlen in den Operatoren der grünen Blöcke mit einer Variablen definiert werden.

Unter dem letzten Bereich eigener Blöcke können selbstständig Blöcke erstellt und benannt werden, indem Beispielsweise ein individueller Block mit der Funktionsweise eines Fühlen-Blocks erstellt wird.

Um Blöcke miteinander zu verknüpfen, müssen diese in den vorhergesehenen Bereich eingefügt werden. Welche Blöcke an welcher Stelle verknüpft werden können ist an der Blockform zu erkennen (Abbildung 4). Die Verbindung von Blöcken geschieht dabei in vertikaler Ausrichtung. Um Blöcke wieder zu entfernen müssen diese in den linken Auswahlbereich geschoben werden, in dem diese dann durch Loslassen des Blocks verschwinden.

Blockform	Blockbeschreibung
	Kopf-Blöcke können ein Skript starten. Der obere Teil ist leicht abgerundet. Unten kann ein Stapel-Block angefügt werden.
	Stapel-Blöcke führen die wesentlichen Anweisungen aus. Oben befindet sich eine Einbuchtung und unten eine Ausbuchtung, an denen Blöcke angefügt werden können.
	Klammer-Blöcke definieren eine Bedingung oder präzisieren eine Ausführung. Anhand der Ausbuchtungen ist zu erkennen, an welcher Stelle Stapelblöcke eingefügt werden können. Im Blockbeispiel kann zudem ein Wahrheitsblock eingefügt werden.
	Wahrheits-Blöcke erfragen einen Zustand. Sie sind sechseckig und können innerhalb bestimmter Blöcke in die passende sechseckige Umrandung eingefügt werden.
	Wert-Blöcke enthalten Zahlen oder Zeichen. Sie sind rund und können innerhalb bestimmter Blöcke in die passende runde Umrandung eingefügt werden.
	Abschluss-Blöcke beenden ein Skript. Sie enthalten keine Ausbuchtung unten, sodass keine weiteren Blöcke angefügt werden können.

Abbildung 4: Blockformen illustriert anhand von Beispielen und dessen Funktionen.²⁶

Die ausgeführten Anweisungen können im rechten oberen Bereich beobachtet werden, sodass direkt kontrolliert werden kann, ob der Algorithmus das ausführt, was tatsächlich geplant war. Zudem ist es in diesem Areal möglich, die Figuren an eine beliebige Position zu verschieben. Im rechten unteren Bereich besteht die Möglichkeit Figuren und Hintergründe hinzuzufügen. Dabei kann aus einer Auswahl von 339 Figuren sowie 85 Hintergründen ausgewählt werden, welche von Scratch bereitgestellt werden. Zudem besteht die Möglichkeit Figuren und Hintergründe zu bearbeiten sowie eigene Darstellungen hochzuladen.

²⁶ Grafik selbst erstellt. Abbildungen der Blöcke aus Scratch Foundation, 2021, online unter: <https://scratch.mit.edu/projects/editor/?tutorial=getStarted>. (Stand 16.12.21), Blockbezeichnungen aus <https://de.scratch-wiki.info/wiki/Blockformen>.

Über die Registerkarte Datei können Projekte gespeichert sowie hochgeladen werden. Weiterhin besteht die Möglichkeit Projekte anderer Nutzer auszuführen, sowie dessen Skript hochzuladen und zu bearbeiten. Die von Scratch angebotenen Anleitungen können benutzerfreundlich eingebunden werden. Über den Reiter Tutorien können diese ausgewählt werden, wodurch nach Auswahl einer Anleitung ein zusätzliches Fenster auf der Bearbeitungsseite geöffnet wird. Dadurch wird vom Scratch-Team ermöglicht, dass Anleitungsschritte direkt in Scratch durchgeführt werden können, ohne dass das Browserfenster gewechselt werden muss.

2.4) ScratchJr

2.4.1) Allgemeines

ScratchJr ist eine von Scratch inspirierte Applikation (App), welche eine jüngere Zielgruppe von Kindern im Alter von fünf bis sieben Jahren ins Auge fasst und durch die Lifelong Kindergarten Group des MIT Media Lab, der DevTech Research Group der Tufts University sowie der Playful Invention Company entwickelt und 2014 veröffentlicht wurde.²⁷ Die Entwickler beschreiben die Lernziele der Kinder wie folgt:

Während jüngere Kinder mit ScratchJr programmieren lernen sie, sich mit Hilfe des Computers selbst auszudrücken und kreativ zu sein, [...] Probleme zu lösen, Projekte zu gestalten und sie entwickeln ein Verständnis für logische Abfolgen. [...] Sie benutzen Mathematik und Sprache in einem sinnvollen und motivierenden Kontext und fördern dadurch die Entwicklung frühkindlicher Rechen- und Lesefähigkeit.²⁸

Die App ist kostenfrei für die Betriebssysteme Android und iOS verfügbar und wurde im Jahr 2020 ca. 3 Millionen Mal heruntergeladen.²⁹

Für ScratchJr werden ebenfalls Informationen für den Schul- bzw. Lehrbereich angeboten. Neben Anleitungen zu Aktivitäten, welche in der App durchführbar sind und zum Erlernen der Funktionen dienen³⁰, werden Materialien für Lehrkräfte angeboten. Vergleichbar mit der Seite *ScratchEd* für das Browserprogramm Scratch gibt es die Seite *ScratchJr Connect* für die Applikation. Über diese Website werden Inhalte angeboten, die im Lehrbereich von Nutzen sein können. Nutzer können Ressourcen von anderen Nutzern abrufen, welche sich unter anderem auf den Schulbereich spezialisieren. Projekte können z.B. nach Alter und Schulfach gefiltert werden,

²⁷ Vgl. DevTech Research Group der Tufts University et al., (2021), online unter: <https://www.scratchjr.org/about/info> (Stand 16.12.21).

²⁸ S. ebd. (Stand 16.12.21).

²⁹ Vgl. Scratch Foundation, 2021, online unter: <https://scratch.mit.edu/annual-report> (Stand 04.12.21).

³⁰ Vgl. DevTech Research Group der Tufts University et al., (2021), online unter: <https://www.scratchjr.org/teach/activities> (Stand 16.12.21).

sodass entsprechende Projekte angezeigt und im Unterricht verwendet werden können. Für den Mathematikunterricht der Primarstufe gibt es beispielsweise Projekte, bei denen das Zählverständnis von Schülern gefördert wird. Die aufgelisteten Ideen werden zudem von der DevTech Company auf einer Skala von 1 bis 3 in den Kategorien Kommunikation, Zusammenarbeit, Gruppenbildung, Inhaltserstellung, Kreativität und Auswahlmöglichkeiten bewertet.³¹

2.4.2) Aufbau

Der Bearbeitungsbereich von ScratchJr kann sich ebenfalls grob in drei Bereiche aufteilen lassen. Der untere Bereich ist für die Programmierungen vorgesehen und enthält ähnlich wie Scratch Blöcke zum Implementieren von Anweisungen. Im zentralen Bereich befinden sich die ausgewählten und hinzugefügten Figuren bzw. Hintergründe. Außerhalb dieser Bereiche befinden sich unter anderem Möglichkeiten zum Hinzufügen von Figuren (links), Hintergründen (oben) oder neuer Szenen (rechts).



Abbildung 5: Screenshot der App ScratchJr mit einer Beispielprogrammierung.

³¹ Vgl. DevTech Research Group der Tufts University et al., (2021), online unter: <https://scratchjrconnect.tufts.edu/browse/> (Stand 16.12.21).

Darüber hinaus gelangt man über den ScratchJr Schriftzug im oberen linken Bereich zur Startseite. Hier werden die gespeicherten Projekte aufgelistet und können wieder abgerufen werden. Zudem gibt es auf der Startseite Bereiche für Einführungsvideos und Beispielprojekte, Spracheinstellungen sowie Anleitungen für die Benutzeroberfläche, den Maeditor und die Blöcke der App.

In ScratchJr gibt es insgesamt 26 Blöcke aus sechs Kategorien, welche ähnlich wie im Browserprogramm nach Funktionen farblich kategorisiert sind. Im Gegensatz zum Browserprogramm umfasst ScratchJr nur eine Blockform, welche der Form eines Puzzlestücks ähnelt (siehe Abbildung 5 unten). Die gelben Blöcke enthalten auslösende Elemente, durch welche ein Skript gestartet wird. Bewegungsblöcke sind blau und lassen Figuren beispielsweise in bestimmte Richtungen bewegen. Aussehensblöcke sind lila und verändern das Erscheinungsbild von Figuren, während grüne Klangblöcke vorher aufgenommene Töne abspielen lassen. Steuerungsblöcke können Anweisungen stoppen, wiederholen oder verlangsamen bzw. beschleunigen. Die letzte Kategorie der Abschlussblöcke enthält Elemente, welche das Ende eines Algorithmus bestimmen können und z.B. den Wechsel zu einer anderen Seite ausführen.

2.4.3) Handhabung

Um Blöcke zum Skript hinzuzufügen, müssen diese in den unteren Bearbeitungsbereich gezogen werden. In diesem Bereich können die Blöcke miteinander verknüpft werden können, indem diese horizontal aneinander gereiht werden. Um Blöcke zu entfernen können diese in den Blockauswahlbereich geschoben werden, in welchem diese dann verschwinden.

Bei der Figurauswahl kann aus 120 bestehenden Figuren ausgewählt werden, während es 28



Hintergründe gibt. Sowohl die Figuren als auch Hintergründe können zudem bearbeitet werden und es besteht die Möglichkeit auf leeren Seiten selbstständig Illustrationen zu erstellen. Außerdem ist es möglich Fotos per Kamera hochzuladen und diese für die Arbeit mit ScratchJr zu verwenden.

Weiterhin gibt es eine Ansichtsfunktion, mit der ein kariertes Rastergitter (Abbildung 6) über dem zentralen Projektbereich erscheint. Anhand dieses Gitters kann

Abbildung 6: Gitteransicht in ScratchJr.

nachvollzogen werden, um welche Distanz Figuren z.B. eine Strecke gehen.

2.5) Scratch und ScratchJr im Vergleich

Abbildung 7 stellt die beiden Programme in den wesentlichen Aspekten gegenüber, um einen direkten Vergleich in übersichtlicher Form zu bieten. Eine Gegenüberstellung der Blöcke von Scratch und ScratchJr befindet sich im Anhang 4.

	Programmvariante	
	Scratch	ScratchJr
Erscheinungsjahr	2007	2014
Programmart	Browserprogramm	Applikation
Systemvoraussetzungen	Chrome (63+) Edge (15+) Firefox (57+) Safari (11+)	iOS 8+ Android 4.2
Anzahl Block-Kategorien	9	6
Anzahl Blöcke insgesamt	107	26
Anzahl Blockformen	6	1
Blockausrichtung	vertikal	horizontal
Anzahl Figuren	339	120
Anzahl Hintergründe	85	28
Alter Zielgruppe	8-16 Jahre	5-7 Jahre
Kosten	kostenlos	kostenlos

Abbildung 7: Vergleich der Programmvarianten Scratch und ScratchJr.

3.) Kompetenzerwerb mit Hilfe von Scratch

Mit Scratch können wichtige Kompetenzen von Schülern aus verschiedenen Bereichen gefördert werden. Im Fach Mathematik gibt es bereits einige Studien, welche auf eine positive Korrelation hinweisen. Darüber hinaus zeigt Scratch in weiteren Bereichen Potenziale auf, wie z.B. einen möglichen Erwerb informatischer Kompetenzen. In diesem Kapitel werden diese Potenziale anhand der von der GI publizierten Kompetenzen für informatische Bildung sowie einem Kompetenzbereich des Medienkompetenzrahmens NRW herausgestellt. Außerdem werden ausgewählte Studienergebnisse zu mathematischen Kompetenzen aufgezeigt, um den Nutzen des Programms für den (Mathematik-)Unterricht aufzuzeigen. Zudem wird eine Studie über die Auswirkung von ScratchJr auf den Erwerb spezieller informatischer Kompetenzen aufgezeigt.

3.1) Erwerb informatischer Kompetenzen

Die GI hat seit der Gründung 1969 viele Empfehlungen im Bereich der Informatik publiziert, welche unter anderem auf den Lernbereich Schule beziehen. Die Empfehlungen für informative Bildung im Primarbereich wurden 2019 veröffentlicht und umfassen je fünf Kompetenzen im Prozess- und Inhaltsbereich. Scratch ist dabei insbesondere für den Inhaltsbereich Algorithmen interessant, da sich die grundlegende Arbeit in Scratch - die Entwicklung von Programmiersequenzen – mit Algorithmen vergleichen lassen. Algorithmen definiert die GI als „Handlungsvorschriften bzw. Ablaufbeschreibungen“³², welche durch Scratch entwickelt werden können. Das Entwerfen von Algorithmen sei zu konstruktiv und kreativ.³³ Letztere wird unter anderem dadurch ermöglicht, dass Lösungen oftmals auf unterschiedlichen Wegen erreicht werden können.³⁴ Die Vielfältigkeit von Lösungswegen ist durch Scratch realisierbar, da beispielsweise Problemstellungen oder Vorgaben auf unterschiedliche Weise gelöst werden können.

Die konkret aufgezeigten Kompetenzerwartungen der GI für den Primarbereich im Inhaltsbereich der Algorithmen lassen sich ebenfalls mit Scratch verbinden. Beispielsweise wird erwartet, dass Schüler algorithmische Grundbausteine verwenden oder Grundbausteine von Algorithmen entwickeln, erstellen und testen.³⁵ Diese Grundbausteine können mit den in den beiden Programmvarianten von Scratch verwendeten Programmierblöcken verglichen werden. Konkreter werden diese Grundbausteine von der GI als Anweisung, Sequenz, Wiederholung und Verzweigung beschrieben, welche sowohl durch ScratchJr als auch durch Scratch realisierbar sind.

Eine zusätzliche Kompetenzerwartung beschreibt das Realisieren von Algorithmen in verschiedenen Darstellungsformen.³⁶ Eine Darstellungsform könnte dabei ein in Scratch erstellter Algorithmus sein (siehe auch Kapitel 4.2.3).

Darüber hinaus können durch Scratch Prozessbereiche angesprochen werden. Exemplarisch kann eine Aufgabenstellung dargeboten werden, in welcher eine (Real-)Situation aus der Erfahrungswelt der Kinder durch Scratch in Form einer Geschichte dargestellt werden soll. Hierdurch wird der Prozessbereich des Modellierens und Implementierens angesprochen, welcher

³² S. Gesellschaft für Informatik e.V., 2019, S. 13.

³³ Vgl. ebd. S. 13.

³⁴ Vgl. ebd. S. 10.

³⁵ Vgl. ebd. S. 13.

³⁶ Vgl. ebd. S. 13.

von den Schülern das Erfassen von Situationen (z.B. Realerfahrung im Leben der Schüler) sowie das Entwickeln von informatischen Modellen mit Hilfe von geeigneten Hilfsmitteln (z.B. Scratch) erfordert.³⁷

Weiterhin können die Prozessbereiche des Strukturierens und Vernetzens durch Scratch angesprochen werden, da durch das Programm das Zerlegen von Sachverhalten ermöglicht werden kann, wie es innerhalb dieses Prozessbereichs beschrieben wird.³⁸ Durch das vor Augen führen der einzelnen Bestandteile einer Programmierung mit Hilfe des Verwendens von Blöcken könnte diese Kompetenz gefördert werden.

Das Potenzial von Scratch für den Kompetenzerwerb zeigt sich ebenfalls durch den Medienkompetenzrahmens NRW, welcher in sechs Kompetenzbereiche unterteilt ist. Im sechsten Bereich des Problemlösens und Modellierens finden Algorithmen Einzug, da als Teilkompetenzen das Erkennen und die Bedeutung von Algorithmen beschrieben werden.³⁹ Daneben wird die Kompetenz des Modellierens und Programmierens angeführt, welche wie folgt beschrieben wird: „Probleme formalisiert beschreiben, Problemlösestrategien entwickeln und dazu eine strukturierte, algorithmische Sequenz planen; diese auch durch Programmieren umsetzen und die gefundene Lösungsstrategie beurteilen“.⁴⁰ Den Zugang zu diesem Kompetenzbereich über Scratch wird auch dadurch verdeutlicht, dass die Autoren des Medienkompetenzrahmens NRW das Verwenden des Programms explizit als Unterrichtsidee anführen, um die Kompetenz des Modellierens und Programmierens zu fördern.⁴¹

3.2) Erwerb mathematischer Kompetenzen

Der Blick auf den Lehrplan Mathematik der Primarstufe des Landes NRW zeigt, dass dort der Begriff des Algorithmus bereits Einzug gefunden hat. Im Prozessbereich des Problemlösens wird zum Ende der Klasse 4 erwartet, dass die Schüler Ideen für mögliche Vorgehensweisen entwickeln, wobei sie dabei auch algorithmisch vorgehen sollen.⁴² Darüber hinaus wird im Kompetenzbereich des Argumentierens erwartet, dass die Schüler Vermutungen über

³⁷ Vgl. ebd. S. 8.

³⁸ Vgl. ebd. S. 8.

³⁹ Vgl. Medienberatung NRW et al., 2021, online unter:

https://medienkompetenzrahmen.nrw/fileadmin/pdf/LVR_ZMB_MKR_Rahmen_A4_2020_03_Final.pdf (Stand 16.12.21).

⁴⁰ S. ebd.

⁴¹ Vgl. Medienberatung NRW et al., 2021, online unter:

<https://medienkompetenzrahmen.nrw/unterrichtsmaterialien/detail/scratch-einfuehrung-in-die-blockprogrammierung/> (Stand 21.12.21).

⁴² Vgl. Ministerium für Schule und Bildung des Landes Nordrhein-Westfalen, 2021, S. 82.

algorithmische Muster und Strukturen in der Mathematik anstellen.⁴³ Grundsätzlich zeigen die Prozessbereiche des Lehrplans Mathematik Parallelen mit den Prozessbereichen der GI im Primarbereich auf. Modellieren, Kommunizieren, Argumentieren und Darstellen finden in beiden Kompetenzerwartungen Einzug.⁴⁴ Beispielsweise könnte das Kommunizieren durch das gemeinsame Entwickeln von Programmierungen gefördert werden, indem Vorgehensweisen in der Gruppe nachvollziehbar erläutert werden. Informatische und mathematische Kompetenzen überschneiden sich teilweise, weshalb die Vermittlung von Algorithmen durch Scratch einen Beitrag zur Förderung mathematischer Kompetenzen leisten kann.

Neben den Prozessbereichen werden Algorithmen ebenfalls in den Inhaltsbereichen des Lehrplans Mathematik aufgegriffen. Die Kompetenz des Ziffernrechnens fordert von den Schülern die „einzelnen Rechenschritte der Algorithmen [der Rechenverfahren der Grundrechenarten] an Beispielen in nachvollziehbarer Weise [zu] beschreiben“⁴⁵. Die Vermittlung mathematischer Kompetenzen könnte durch die Verwendung von Scratch mit Hilfe des Aufzeigens algorithmischer Strukturen gelingen.

Konkrete Verwendungsvorschläge für den Mathematikunterricht werden zudem auf der Website *ScratchEd* angeboten. Mit Scratch könnten beispielsweise die Grundlagen des Koordinatensystems, Prozentrechnungen oder geometrische Strukturen erlernt werden, wobei diese den Anforderungsbereich der Grundschule zum Teil übersteigen.⁴⁶

Die Wirksamkeit von Scratch im Mathematikunterricht wurde bereits in einigen Studien untersucht. Die Studie *Developing Mathematical Thinking with Scratch* (2015) von Luis Alberto Calao et al. zeigt einen signifikanten Anstieg im Verstehen mathematischer Prozesse durch die Arbeit mit Scratch.⁴⁷ Bei der Studie wurden die mathematischen Kompetenzen des Modellierens (*modeling*), des logischen Denkens (*reasoning*), des Problemlösens (*problem solving*) und des Übens (*exercising*) von Schülern einer sechsten Klasse in Kolumbien im Zusammenhang mit der Verwendung von Scratch analysiert. Die Klasse wurde dafür in eine Experimental- und Kontrollgruppe aufgeteilt. Die Experimentalgruppe hat dabei über einen Zeitraum von drei Monaten mit Scratch gearbeitet, während die Kontrollgruppe den regulären Mathematikunterricht durchgeführt hat. Vor und nach dem Beginn dieser Intervention wurde ein Pre- bzw. Post-Test

⁴³ Vgl. ebd. S. 84.

⁴⁴ Vgl. ebd. S. 78 sowie Gesellschaft für Informatik e.V., 2019, S. 17.

⁴⁵ S. ebd. S. 88.

⁴⁶ Vgl. Harvard Graduate School of Education, 2016, online unter:

<https://scratched.gse.harvard.edu/resources/scratch-every-subject-math.html> (Stand 01.12.21).

⁴⁷ Vgl. Calao et al., 2015, S. 17.

durchgeführt, anhand derer die mathematischen Kompetenzen der Schüler der Bereiche des Modellierens, logischen Denkens, Problemlösens und Übens festgestellt werden sollten. Die Ergebnisse lassen dabei auf einen positiven Einfluss von Scratch schließen, da in jedem der vier Kompetenzbereiche die Experimentalgruppe einen deutlichen Anstieg verzeichnen konnte, während die Kontrollgruppe nur geringe Verbesserungen bzw. sogar Verschlechterungen aufzeigt.

Zu ähnlichen Ergebnissen kommen Seungki Shin und Phanwoo Park, welche die Auswirkungen von Scratch auf Problemlösefertigkeiten von Grundschülern in Südkorea analysiert hat. Die Ergebnisse der Wissenschaftler zeigen einen signifikanten Unterschied in den Bereichen des Fällens von Entscheidungen (*decision making*), divergenten Denkens (*divergent thinking*) und der Planungsfähigkeit (*planning ability*), welche als Subkategorien des Problemlösens examiniert wurden.⁴⁸

Eine weitere Studie von Olivera Iskrenovic-Momcilovic zeigt einen positiven Effekt von Scratch im Hinblick auf den Lernzuwachs im Bereich der Geometrie.⁴⁹ Vergleichbar mit der Methodik der Studie von Calao et al. wurden 106 Schüler der dritten Klasse einer Grundschule in eine Experimental- und Kontrollgruppe gleicher Größe aufgeteilt. Beide Gruppen lernten im Anschluss grundlegende geometrische Figuren, wobei die Experimentalgruppe im Gegensatz zur Kontrollgruppe Scratch verwendete. Im Anschluss wurde ein Wissenstest durch die Schüler durchgeführt, anhand dessen der Lernerfolg analysiert wurde. Die Resultate zeigen dabei auf, dass es einen signifikanten Unterschied im Erfolg zwischen den beiden Gruppen gibt, bei der die Experimentalgruppe erfolgreichere Ergebnisse erzielte.

Studienergebnisse im Hinblick auf die Verwendung von ScratchJr im Grundschulalter sind weniger aufzufinden, was wahrscheinlich auf das Zielgruppenalter der App zurückzuführen ist. Eine Querschnitts-Studie von Eleni A. Kyza et al. untersuchte unter anderem die Fähigkeiten des *computational thinking* von Kindern beim Verwenden von ScratchJr. Der Begriff des *computational thinking* wird oft im Zusammenhang bei der Arbeit mit Informatiksystem genutzt und kann als „strukturiertes Zerlegen von Problemen wie auch ein konstruktives und kreatives Modellieren von Problemlösungen“⁵⁰ bezeichnet werden. Diese informatischen Kompetenzen lassen sich mit Fertigkeiten der Allgemeinbildung sowie ausgewählten mathematischen

⁴⁸ Vgl. Shin & Park, 2014, S. 119.

⁴⁹ Vgl. Iskenovic-Momcilovic, 2020, S. 1-8.

⁵⁰ S. Gesellschaft für Informatik e.V., 2019, S. 5.

Bereichen vereinbaren und sind somit auch für den Mathematikunterricht von Bedeutung. In der Studie wurden diese Kompetenzen überprüft, indem zwei Kohorten der Altersgruppen von sechs bis sieben und zehn bis zwölf Jahre eine Geschichte über Umweltverschmutzung in ScratchJr programmieren sollten. Diese Geschichten wurden dann dahingehend analysiert, inwiefern sechs Konzepte des *computational thinking* eingeflossen sind.⁵¹ Diese Konzepte umfassten Abstraktion und Problemlösen (*abstraction and problem decomposition*), Parallelität (*parallelism*), Synchronisation (*synchronization*), Ablaufsteuerung (*flow control*), Benutzerinteraktivität (*user interactivity*) und Datenrepräsentation (*data representation*). Die Resultate zeigten auf, dass diese Kompetenzen (bis auf Abstraktion und Problemlösen in der älteren Kohorte) wenig oder nur bis zur Hälfte in den Geschichten wiedergefunden wurden, woraus die Autoren schließen, dass unterschiedliche Herausforderungen bei der Verwendung von ScratchJr eine Ursache dafür sein könnten. Zudem erzielte die ältere Gruppe, bis auf die Konzepte der Ablaufsteuerung und Benutzerinteraktivität bei denen der Anteil gleich groß ist, bessere Ergebnisse, was die Autoren auf die altersgegebenen höheren kognitiven Fertigkeiten zurückführen.

Die aufgezeigten Studienergebnisse deuten auf eine positive Korrelation von der Verwendung von Scratch und mathematischen Kompetenzen hin, während die Arbeit mit ScratchJr keine Hinweise für einen positiven Einfluss im Bereich der informatischen Kompetenzen des *computational thinking* aufweist.

4.) Umstieg von ScratchJr zu Scratch

4.1) Rahmenbedingungen

Die Erprobung möglicher Herausforderungen des Übergangs von ScratchJr zu Scratch findet in einer Arbeitsgemeinschaft des Ganztagsprogramms einer Grundschule statt. Innerhalb dieser AG verwenden Schüler der dritten und vierten Klassenstufe die Programme ScratchJr und Scratch mit Hilfe eines Tablets. Die AG findet einmal wöchentlich in 90-minütigen Sitzungen statt. An der AG nehmen fünf männliche Schüler (ein Drittklässler und vier Viertklässler) sowie neun weibliche Schülerinnen (vier Drittklässlerinnen und fünf Viertklässlerinnen) teil. Alle Teilnehmer haben die beiden Programme vorab nicht verwenden.

⁵¹ Vgl. Kyza et al., 2021, S. 22-30.

4.2) Methodik

Um einen Beitrag dazu zu leisten, Herausforderungen des Übergangs von ScratchJr zu Scratch herauszustellen, werden verschiedene Hypothesen aufgestellt. Diese Hypothesen sollen dann anhand verschiedener Aufgabenstellungen und Fragen überprüft werden. Aus den Ergebnissen sollen dann Rückschlüsse auf mögliche Herausforderungen des Übergangs gezogen werden. Im Rahmen dieser Arbeit wurden dafür die folgenden drei Hypothesen aufgestellt:

1. Die vorherige Nutzung von ScratchJr erschwert den Einstieg in Scratch.

Obwohl die beiden Programme im Grundsatz auf der gleichen Programmiersprache basieren, gibt es einige Unterschiede zwischen den beiden Versionen. Der strukturelle Aufbau, Blockausrichtung und Blockfarben, Auswahlmöglichkeiten, Handling und Umfang zeigen zum Teil große Diskrepanzen auf. Die Auswertungen der Arbeitsaufträge sollen mögliche Herausforderungen beim Übergang von ScratchJr zu Scratch erörtert werden, welche auf die grundsätzlichen strukturellen Unterschiede zurückzuführen sind.

2. Die vorherige Nutzung von ScratchJr fördert die erfolgreiche Bearbeitung mathematischer Inhalte in Scratch.

Wie bereits in Kapitel 3.2) aufgezeigt, deuten Studien auf eine positive Korrelation von Scratch und mathematischen Kompetenzen hin. Zudem zeigt das Programm Potenziale für das Erlernen weiterer informatischer Kompetenzen, während keine Signifikanz beim Erlernen von *computational thinking* im Grundschulalter zu erkennen ist. Da ScratchJr ebenfalls mathematische Bereiche vermitteln kann, wird die Hypothese aufgestellt, dass die Bearbeitung mathematischer Inhalte in Scratch von der vorherigen Nutzung von ScratchJr profitiert. Die Ergebnisanalyse soll Hinweise für mögliche Übergangsherausforderungen beim Behandeln mathematischer Inhalte liefern.

3. ScratchJr ist nicht für den Einsatz in der Grundschule geeignet.

Mit fünf bis sieben Jahren liegt das Alter der Zielgruppe von ScratchJr nur begrenzt im Grundschulalter. Da in der Grundschule dieses Alter größtenteils überschritten ist, wird die Annahme aufgestellt, dass der Einsatz von ScratchJr für den Einsatz in der Primarstufe ungeeignet ist. Herausforderungen des Übergangs von ScratchJr zu Scratch sollen nach Analyse

der Ergebnisse aus der grundsätzlichen Eignung der App für die Grundschule geschlossen werden.

Die drei Hypothesen werden anhand verschiedener Aufgabenstellungen und Fragebögen überprüft. Die Fragen und Aufgaben werden altersgerecht formuliert, um Verständlichkeitsprobleme zu minimieren. Um dies zu gewährleisten, wurden alle Aufgabenstellungen sowie Fragebögen von zwei Lehrerinnen auf Verständlichkeit überprüft und überarbeitet. Beide Lehrkräfte sind jeweils über 20 Jahre im Grundschulbereich tätig und Fachkräfte für das Fach Deutsch. Zudem werden die Aufgaben und Fragen vor Bearbeitung über eine Dokumentenkamera bzw. das Smartboard an der Tafel besprochen und Rückfragen geklärt. Die Bearbeitung der Arbeitsaufträge wird in Einzelarbeit durchgeführt. Bei Bearbeitungsproblemen der Arbeitsaufträge werden Hilfestellungen gegeben, sofern diese nicht zur Lösung der Aufgaben beitragen (z.B. Probleme durch Fehlermeldungen auf dem Tablet).

4.2.1) Erste Hypothese

Die erste Hypothese wird mit Hilfe einer Aufgabenstellung sowie eines Fragebogens überprüft (siehe Anhang 1). Bei der Bearbeitung der Aufgabe sollen dabei fünf Bedingungen erfüllt werden, welche sich auf die Kompetenzerwartungen der Empfehlungen der GI beziehen. Dort wird im Bereich der Algorithmen zum Ende der vierten Klassen von den Schülern erwartet, dass diese „Algorithmen mit den algorithmischen Grundbausteinen Anweisung, Sequenz, Wiederholung und Verzweigung“⁵² entwerfen. Die ersten drei Vorgaben erfordern dabei das einfache Ausführen von Anweisungen bzw. Sequenzen, Aufgabe vier die Implementierung einer Wiederholung und Aufgabe fünf das Erarbeiten einer Verzweigung. Diese Anforderungen wurden so gewählt, dass diese auch in der vereinfachten ScratchJr Version durchführbar sind. Außerdem wurde sichergestellt, dass alle Schüler diese Vorgaben vor der Bearbeitung im Browserprogramm in ScratchJr korrekt umgesetzt haben. Dadurch kann überprüft werden, ob die Kinder relativ gängige und einfache Anforderungen im Browserprogramm durchführen können oder Probleme bei der Bearbeitung auftreten. Die Aufgabe wird den Kindern bei der ersten Verwendung des Browserprogramms Scratch gestellt. Damit das Risiko minimiert wird, dass die Kinder mit dem neuen Aufbau des Programms überfordert sind und deshalb keine Vorgabe lösen, wird vor dem Durchführen des Arbeitsauftrags die Programmstruktur am Smartboard kurz

⁵² S. Gesellschaft für Informatik e.V., 2019, S. 13.

erläutert. Um keine Hilfestellungen für die Lösung der Aufgaben zu leisten, werden dabei nur die drei bereits in Kapitel 2.1) angesprochenen Bereiche des Programms kurz aufgezeigt. Der Zeitrahmen für die Bearbeitung der Aufgabe wird von den Beobachtungen der Kinder bei der Verwendung von ScratchJr abgeleitet. Um die gleichen Vorgaben des Arbeitsauftrages in ScratchJr erfolgreich umzusetzen benötigen die Schüler ca. fünf bis zehn Minuten. Da Scratch eine deutlich höhere Anzahl an Blöcken vorweist und dadurch Zeiteinbußen durch Suchen geeigneter Blöcke entstehen können, wird den Schülern für die Aufgabe ein Zeitrahmen von 15 Minuten gewährt.

Darüber hinaus wird nach der Bearbeitung des Arbeitsauftrags ein Fragebogen ausgefüllt, welcher sich auf die fünf Vorgaben bezieht. Die Kinder beantworten, inwiefern das Einhalten der Vorgaben zu Problemen geführt hat. Dabei können vier Werte (sehr leicht, leicht, schwer, sehr schwer) angegeben werden. Zudem kann in einem Freitextfeld beschrieben werden, weshalb mögliche Probleme aufgetreten sind.

Bei der Auswertung der Ergebnisse werden die Anzahl erfolgreich durchgeföhrter Vorgaben der Anzahl fehlerhaften oder nicht durchgeföhrten Vorgaben gegenübergestellt. Ebenso werden die Antworten des Fragebogens prozentual aufbereitet, um das Schwierigkeitsempfinden darzustellen. Anhand der Freitextantworten sollen konkrete Gründe für mögliche Schwierigkeiten bei der Aufgabenstellung herausgearbeitet werden.

4.2.2) Zweite Hypothese

Die zweite Hypothese wird mit Hilfe von zwei Aufgaben überprüft, wobei arithmetische und geometrische Inhaltskompetenzen angesprochen werden. Die Aufgaben leiten sich daraus ab, welche Kompetenzen durch die vorherige Arbeit mit ScratchJr theoretisch vermittelt werden könnten. Die Aufgaben werden in der dritten Sitzung nach dem Umstieg auf Scratch bearbeitet, sodass die Schüler mit der Struktur und Handhabung des Browserprogramms vertraut sind. Dadurch soll gewährleistet werden, dass der Fokus bei der Aufgabenbearbeitung auf den mathematischen Kompetenzen liegt und Probleme mit dem Handling des Programms minimiert werden.

In der ersten Aufgabe wird den Schülern eine unvollständige Programmierung zur Verfügung gestellt, welche anschließend vervollständigt werden soll (Abbildung 8). Die Aufgabenstellung (siehe Anhang 2) fordert, dass sich die Figur einer Katze in die Richtungen links, rechts, oben und unten drehen kann. Außerdem soll die Katze drei verschiedene Entferungen (kurz, mittel

und lang) laufen können, wobei die Schrittzahl immer durch sieben teilbar sein soll. Diese Anweisungen sollen per Druck auf den dafür vorhergesehenen Knopf durchgeführt werden. Vor dem Austeilen des Arbeitsauftrags wird am Smartboard exemplarisch gezeigt, inwiefern die Knöpfe keine oder fehlerhafte Anweisungen ausführen. Anschließend wird Zeit für Rückfragen geboten, um das Risiko zu vermindern, dass der Arbeitsauftrag missverstanden wird. Die Programmierung ist dabei so weit vervollständigt, dass die einzelnen Richtungs- und Entfernungsknöpfe nicht bearbeitet werden müssen, sodass nur bei den Algorithmen der Katzenfigur Änderungen vorgenommen müssen. Dadurch soll sich auf die Nutzung mathematischer Kompetenzen zur Lösung der Aufgabe fokussiert werden.

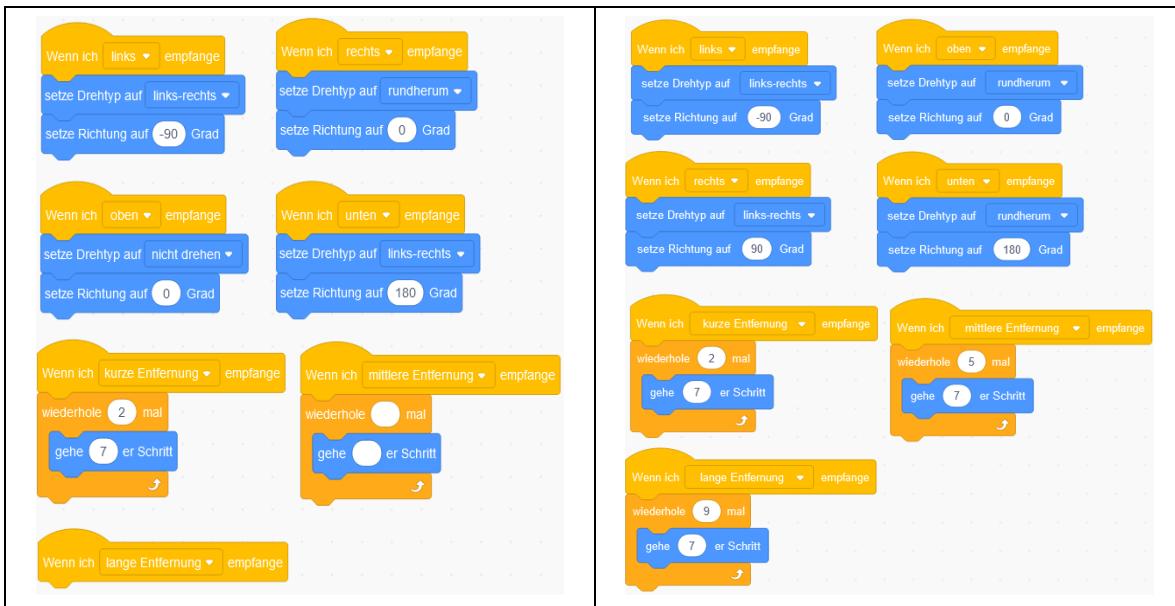


Abbildung 8: Von den Schülern zu vervollständigende Programmierung (links) und eine mögliche Lösung des Arbeitsauftrags (rechts).

Die geforderten Kompetenzbereiche zur erfolgreichen Beantwortung der ersten Aufgabe liegen einerseits im Inhaltsbereich Raum und Form des Lehrplans Mathematik für die Primarstufe. Der Lehrplan fordert zum Ende der Schuleingangsphase, dass Schüler in der Lage sein sollten, „Wege und Lagebeziehungen (u.a. rechts, links, über, unter, hinter, vor) zwischen konkreten und bildlich dargestellten Gegenständen [zu beschreiben]“⁵³. Für die Drehbewegungen wird von den Schülern das angeben von Gradzahlen gefordert, welche die Lernbereiche der Grundschule übersteigen. Bei verändern dieser Zahl wird allerdings zudem die Richtung anhand einer Scheibenabbildung verdeutlicht (Abbildung 9), sodass die Richtungsveränderung ohne die Kenntnis über Funktionen von Gradzahlen erkannt werden kann. Der Richtungspfeil kann dabei manuell in die gewünschte Richtung gezogen werden, wobei sich die Gradzahl automatisch mitverändert. Neben der Gradanzeige ist es zudem notwendig, den richtigen Drehtyp auszuwählen (siehe Abbildung 8), da die Katze bei falscher Auswahl nicht in die jeweilige Richtung schaut. Durch diese Anforderungen wird die Kompetenz „die Lage von ebenen Figuren und Körpern in der Vorstellung [zu verändern] [...] (u.a. Kippbewegungen)“⁵⁴ des Inhaltsbereiches Raumorientierung und Raumvorstellung zu Ende der vierten Klasse gefördert, da sich die Kippbewegungen der Katze vorgestellt werden und in Verbindung mit dem richtigen Drehtyp (z.B. rundherum) gebracht werden müssen.

Die Bearbeitung der Entfernungsanweisungen fordert Operationsverständnis innerhalb des Inhaltsbereiches der Multiplikation. Dort wird von den Schülern erwartet, dass „Situationen des Wiederholens, Zusammenfassens, Vergleichens und Multiplikationsaufgaben einander zu[geordnet]“⁵⁵ werden. Die Wiederholungsblöcke von Scratch bieten dabei einen Zugang zu diesem Bereich. Durch die Vorgabe, dass die Schrittentferungen durch sieben teilbar sein müssen, sollen die Schüler Wiederholungen zu der Schrittzahl sieben anwenden. Dabei müssen die Kinder beachten, dass die mittlere Distanz zwischen der kurzen und langen Distanz liegt. Bei der zu behandelten Programmierung ist die kurze Distanz bereits korrekt mit dem Wiederholungs- und Gehe-Block und den Werten zwei und sieben vorgegeben. Bei der mittleren

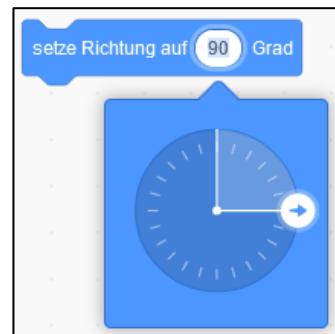


Abbildung 9: Scheibengrafik zum Bestimmen der Drehrichtung.

⁵³ S. Ministerium für Schule und Bildung des Landes Nordrhein-Westfalen, 2021, S. 89.

⁵⁴ S. ebd. S. 89.

⁵⁵ S. ebd. S. 86.

Distanz sind nur die Blöcke ohne Zahlen voreingestellt und bei der langen Distanz nur das Ereignis *lange Entfernung* (siehe Abbildung 8).

Bei der zweiten Aufgabe wird wie in der Aufgabenstellung davor eine unvollständige bzw. teilweise fehlerhafte Programmierung aufgezeigt, welche korrekt vervollständigt werden soll (Abbildung 10). Gleichermaßen wie in der ersten Aufgabe wird das fehlerhafte Verhalten der jeweiligen Knöpfe am Smartboard präsentiert und die Möglichkeit für Rückfragen gegeben. In dieser Aufgabe sind drei geometrische Figuren gegeben (Rechteck, Quadrat, Dreieck), welche sich mit Betätigung des jeweiligen Bedienungselements vergrößern bzw. verkleinern sollen (Abbildung 10).⁵⁶ Als Hilfsmittel wurde der Hintergrund in Form einer Gitteransicht gewählt, welche sich mit einem karierten Blatt vergleichen lässt. Weiterhin wird ein Zurücksetzen-Knopf eingebettet, sodass die Schüler bei Problemen mit der Aufgabe die Figuren auf die Originalgröße zurücksetzen können. Im Gegensatz zur ersten Aufgabe wird von den Schülern gefordert, nicht nur bei einer Figur Algorithmen zu bearbeiten, sondern bei den drei geometrischen Figuren Änderungen vorzunehmen.

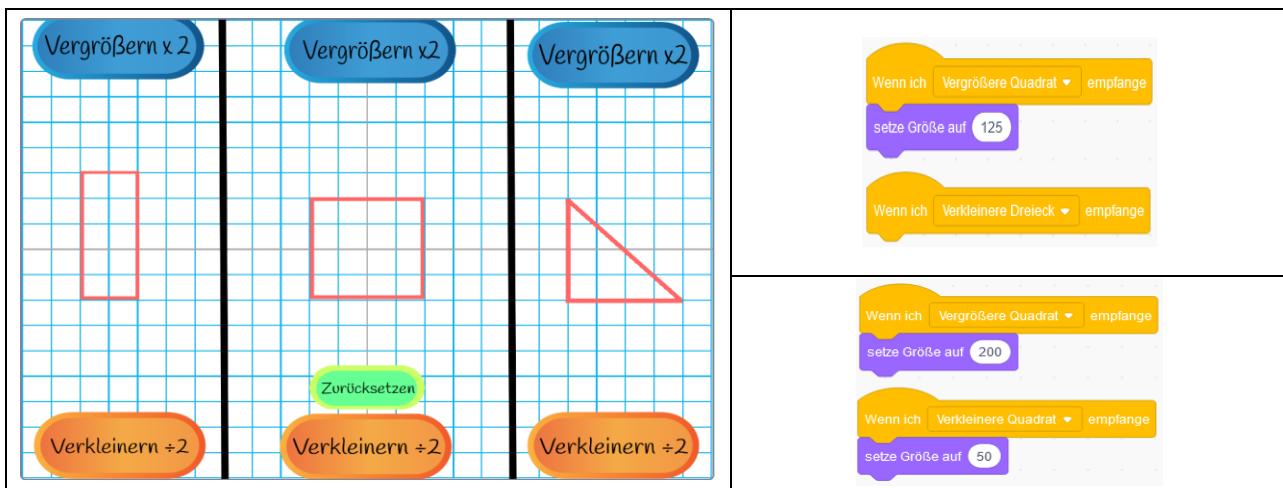


Abbildung 10: Links: Ansicht der Programmierung mit den geometrischen Figuren Rechteck, Quadrat, Dreieck sowie den ausführenden Knöpfen. Rechts: Ausschnitt eines zu vervollständigenden Algorithmus für das Quadrat (oben) und eine mögliche Lösung (unten).

Für die richtigen Programmierungen wird von den Schülern einerseits aus dem Inhaltsbereich der Ebenen Figuren am Ende der Schuleingangsphase die Kompetenz gefordert die „geometrischen Grundformen (Rechteck, Quadrat, Dreieck [...]]) [zu identifizieren]⁵⁷. Diese Anforderung wird

⁵⁶ Die Verdopplung bzw. Verkleinerung muss nur einmal im Vergleich zum Originalbild durchgeführt werden, da die mehrfache korrekte Größenveränderung eine Programmierung erfordert, welche die Kompetenzen der Schüler zum Bearbeitungszeitpunkt der Aufgabe deutlich übersteigt.

⁵⁷ S. Ministerium für Schule und Bildung des Landes Nordrhein-Westfalen, 2021, S. 90.

dadurch angesprochen, dass einige Buttons für die Größenveränderungen die falschen Figuren vergrößern bzw. verkleinern (siehe Abbildung 10 rechts). Die Schüler müssen erkennen, welche Figuren das Rechteck, Quadrat und Dreieck sind und die richtige Zuordnung in der Programmierung wählen.

Weiterhin werden am Ende der Klasse 4 das Vergleichen von Flächeninhalt sowie das maßstäbliche Verkleinern und Vergrößern ebener Figuren erwartet.⁵⁸ Da letztere Kompetenzen nach Rücksprache mit den Klassenlehrerinnen der Schüler noch nicht im Mathematikunterricht behandelt wurden, ist in der Aufgabenstellung von einer ungefähren Verdoppelung bzw. Verkleinerung der Figuren und nicht des Flächeninhalts die Rede (siehe Anhang 2). Trotz der noch nicht behandelten Inhalte zu diesen Kompetenzerwartungen könnte eine erfolgreiche Bearbeitung dieser Aufgabe auf ein Verständnis von Größenveränderungen mit Bezug zum Flächeninhalt geschlossen werden, welche auf die Arbeit mit ScratchJr zurückzuführen sein könnte.

Neben den Kompetenzen im Bereich von Raum und Form ist zudem ein Zahlenverständnis der Kinder von Nöten, da in diesem Kompetenzbereich Beziehungen zwischen Zahlen mit konkretem Hinweis auf die Hälfte und das Doppelte von Zahlen erwartet werden.⁵⁹ Da die Normalgröße der Figuren standardmäßig mit dem Wert von 100 eingestellt ist, können die Größenänderungen in den Algorithmen auf zwei Weisen durchgeführt werden. Die Größenveränderungen können durch die Blöcke *Ändere Größe um* oder *Ändere Größe auf* durchgeführt werden. Für die korrekte Größenänderung um einen Wert müssen die Zahlen 100 für die Vergrößerung und -50 für die Verkleinerung eingetragen werden, während bei der Größenveränderung auf einen Wert die Zahlen 200 für die Vergrößerung und 50 bei der Verkleinerung eingestellt werden müssen. Die Kompetenz des Zahlverständnisses wird mit dieser Aufgabenstellung angesprochen, da diese Zusammenhänge von den Schülern erkannt werden müssen.

Diese beiden Aufgabenteile sowie der Fokus auf den angegebenen Kompetenzerwartungen sind zudem von den Möglichkeiten bei ScratchJr abgeleitet. Die in Aufgabe 1 erforderten Richtungs- und Entfernungsbewegungen sind auch mit ScratchJr umsetzbar, werden jedoch unterschiedlich realisiert. Im Gegensatz zum Browserprogramm werden keine Drehtypen und Gradwerte für die Drehbewegungen angegeben, sondern Drehblöcke dafür verwendet, wodurch sich die Figur nach links und rechts drehen kann (siehe Anhang 4). Die Bewegungen für rechts, links, oben und

⁵⁸ Vgl. ebd. S. 90.

⁵⁹ Vgl. ebd. S. 86.

unten werden in ScratchJr ebenfalls durch eigenständige Blöcke realisiert. Darüber hinaus sind Wiederholungsblöcke in beiden Programmvarianten vorhanden, welche für die Entfernungsangaben genutzt werden sollen.

Die zweite Aufgabe der Größenveränderungen ebener Figuren können ebenfalls von der App abgeleitet werden. Hierfür gibt es jeweils einen Block für die Verkleinerung und Vergrößerung von Figuren. Dabei beschreibt der eingetragene Wert den Grad der Größenänderung, wobei der höchste Wert mit 99 begrenzt ist. Zudem kann das Verwenden der Gitteransicht in ScratchJr mit dem karierten Hintergrundbild in der zweiten Aufgabe verglichen werden.

Die Ergebnisse der Schüler werden abschließend anhand der Anzahl der behobenen Fehler aufgearbeitet. Diese werden durch die angesprochenen Kompetenzen differenziert aufgearbeitet und in die Kategorien Raum und Form (Raumorientierung und Raumverständnis, Ebene Figuren) sowie Zahlen und Operationen (Operationsverständnis, Zahlverständnis) aufgeteilt.

4.2.3) Dritte Hypothese

Um der Annahme der dritten Hypothese nachzugehen, werden den Schülern eine kurze Aufgabenstellung sowie ein Fragebogen (siehe Anhang 3) ausgeteilt.

Die Aufgabe und Fragen dieses Arbeitsauftrages wurden vor dem Übergang zu Scratch bearbeitet, damit die Informationen daraus auf ScratchJr zurückzuführen sind. Zum Zeitpunkt der Bearbeitung haben die Kinder in insgesamt acht Sitzungen ScratchJr verwendet.

Die Aufgabe fordert von den Schülern, dass sie ein Flussdiagramm in ScratchJr übertragen. Im Vorhinein wurden den Kindern die Struktur sowie Funktionen verwendeter Bausteine innerhalb eines Flussdiagramms erläutert. Neben dieser Aufgabe wird zudem ein Fragebogen ausgefüllt, welcher sechs Aussagen umfasst. Die erste Frage soll erörtern, inwiefern die Arbeit mit ScratchJr über- oder unterfordert, indem das Arbeiten mit ScratchJr als sehr leicht, leicht, schwer oder sehr schwer beschrieben wird. Für die Aussagen zwei bis sechs können die Schüler vier Zustimmungsgrade (stimme zu, stimme eher zu, stimme eher nicht zu, stimme nicht zu) auswählen. Die Aussagen zwei (Ich habe das Gefühl, dass ich noch Vieles in ScratchJr entdecken kann.), drei (Mir fallen keine neuen Ideen ein, die ich mit ScratchJr durchführen kann) und vier (Ich wünschte mir, dass ich mehr Auswahlmöglichkeiten bei ScratchJr hätte) sollen Aufschluss darüber geben, inwiefern der Umfang der App für die Grundschüler ausreichend ist. Aussage fünf (Seitdem ich mit ScratchJr arbeite, habe ich Vieles dazu gelernt.) dient dazu, einen möglichen Lernzuwachs der Kinder herauszuarbeiten. Die letzte Frage zielt darauf ab, womit sich die Kinder

bei der Verwendung der App die meiste Zeit befassen. Bei dieser Frage werden den Kindern vier Antwortmöglichkeiten (Gestalten von Figuren und Hintergründen, Programmieren von Geschichten, Programmieren von Spielen, Entdecken/Ausprobieren neuer Möglichkeiten) sowie ein Freitextfeld für Sonstiges vorgegeben. Die vorgegebenen Antwortmöglichkeiten sind dabei von den Beobachtungen in den bis dahin stattgefundenen Sitzungen in der AG abgeleitet.

Die Analyse des Arbeitsauftrags soll überprüfen, ob die Arbeit mit Scratch Kompetenzen des Inhaltsbereichs der Algorithmen vermittelt. Die Aufgabe fußt auf den Kompetenzerwartungen der GI welche am Ende der Klasse 4 von Schülern erwartet, Algorithmen in unterschiedlichen Darstellungsformen zu realisieren.⁶⁰ Hierfür besteht die Aufgabe darin, ein Flussdiagramm in ScratchJr zu „übersetzen“. Bei der Auswertung der Aufgabe werden die Fehlerpunkte der einzelnen Lösungen aufgelistet, woraus die Erfolgsquote geschlossen wird. Anhand der Fehleranzahl können dann Schlüsse geschlossen werden, ob sich die App für die Vermittlung der angesprochenen Kompetenz eignet.

Für die Analyse des Fragebogens werden die drei Bereiche Handhabung (Frage 1), Umfang (Fragen 2 bis 4) und Lernzuwachs (Frage 5) kategorisiert. Je nach Schwierigkeits- (Frage 1) bzw. Zustimmungsgrad (Fragen 2 bis 5) wird anhand dieser Kategorien erörtert, ob die App für die Schüler geeignet ist. Schwierigkeiten beim Umgang oder ein zu geringer Umfang des Programms könnten die Hypothese stützen.

Die Angabe, womit die Schüler die meiste Zeit verbringen soll im weiteren Analyseschritt die Eignung für die Vermittlung informatischer Prozesskompetenzen aufzeigen. Falls die Kinder überwiegend die Antwortmöglichkeiten des Programmierens oder Entdeckens anführen, könnte dies auf die Vermittlung der Kompetenzen hinweisen, da dadurch beispielsweise Prozesskompetenzen wie Modellieren und Implementieren oder Strukturieren und Vernetzen angeeignet werden. Das Gestalten von Figuren oder Hintergründen könnte zwar die Kreativität der Schüler ansprechen, fördert den Erwerb informatischer Kompetenzen jedoch nicht in dem gleichen Ausmaß wie durch das Programmieren von Spielen oder Geschichten.

Darüber hinaus werden die Ergebnisse der anderen Hypothesen mit in die Analyse dieser Annahme einfließen. Wenn die Ergebnisse der ersten beiden Hypothesen auf einen positiven Effekt von ScratchJr hindeuten, könnte auf eine Eignung der App für die Grundschule geschlossen werden, während eine negative Korrelation auf das Gegenteil hinweisen könnte.

⁶⁰ Vgl. Gesellschaft für Informatik e.V., 2019, S. 13.

4.3) Ergebnisse⁶¹

Die Ergebnisse der Aufgaben zur Überprüfung der ersten Hypothese geben ein durchmisches Bild an. Die Vorgaben wurden größtenteils nicht von der Mehrheit der Schüler erfolgreich umgesetzt (Abbildung 11). Die dritte Vorgabe (Figur sagt etwas) wurde mit zehn erfolgreichen gegenüber drei fehlerhaften Programmierungen am besten umgesetzt, während die zweite Vorgabe (Eine Figur vergrößert oder verkleinert sich) von knapp über der Hälfte der Schüler gelöst werden konnte. Die vierte Vorgabe (Eine Anweisung wird wiederholt) konnten lediglich vier Schüler richtig lösen. Die erste (Figur bewegt sich in vier Richtungen) und letzte Vorgabe (Eine Figur verändert ihr Verhalten, sobald diese berührt wird) konnte von keinem Schüler durchgeführt werden konnte.

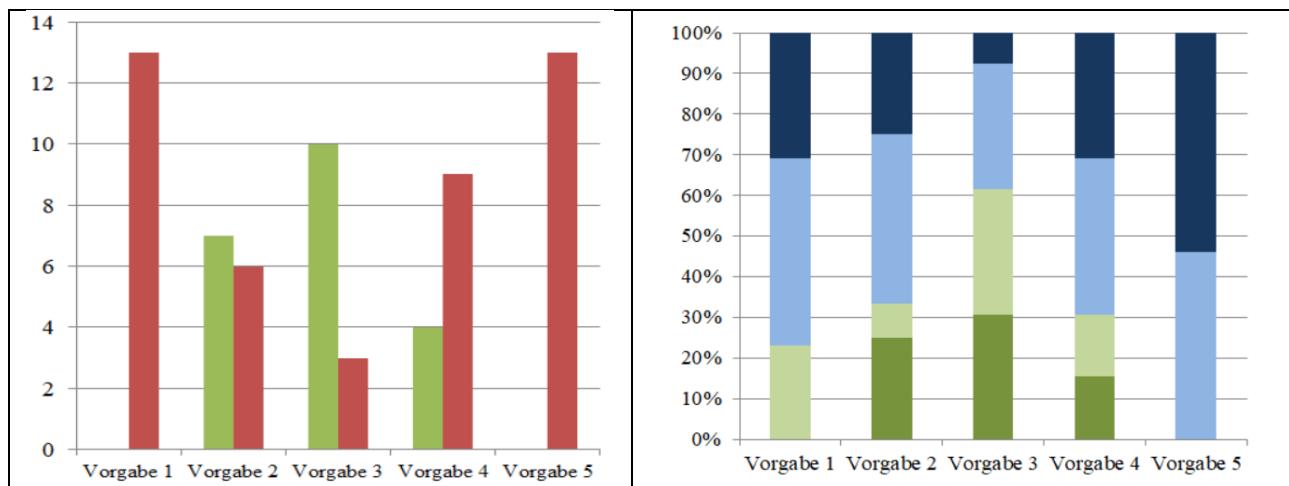


Abbildung 11: Links: Anzahl richtiger Programmierungen (grün) und fehlerhafter Programmierungen (rot) je Vorgabe. Rechts: Empfinden des Schwierigkeitsgrades in Prozent je Vorgabe von sehr leicht (dunkelgrün), leicht (hellgrün), schwer (hellblau) bis sehr schwer (dunkelblau).

Das Empfinden der Kinder, wie leicht bzw. schwer die Bearbeitung der Aufgaben gefallen ist, deckt sich teilweise mit den Ergebnissen der ersten Aufgabe. Bis auf die dritte Vorgabe, bei der die Bearbeitung leichter gefallen ist, bewerten die Kinder die anderen Vorgaben überwiegend als schwer umsetzbar (Abbildung 11). Insbesondere die fünfte Vorgabe ist jedem Schüler schwer gefallen, was sich mit den Ergebnissen deckt, dass kein Kind eine erfolgreiche Programmierung dieser Vorgabe durchgeführt hat.

⁶¹ Krankheitsbedingt haben 13 Schüler an den Arbeitsaufträgen der ersten und dritten Hypothesen sowie 12 Schüler an den Arbeitsaufträgen der zweiten Hypothese teilgenommen. Bedingt durch die Einschränkungen der Corona-Pandemie waren keine nachträglichen Bearbeitungen durch die fehlenden Schüler durchführbar.

Die Auswertung der Ausführungen der Schüler hinsichtlich der Gründe für das Aufkommen von Problemen weist insbesondere zwei Problemkategorien auf. 23 Schüler haben angegeben, dass sie die benötigten Blöcke nicht gefunden haben, während zwölf Mal angeführt wurde, keine Zeit mehr gehabt zu haben. Einmalig wurden die Begründungen abgegeben, dass es zu kompliziert und zu langweilig war.

Die Ergebnisse der Aufgaben hinsichtlich der mathematischen Kompetenzen wurden in die beiden Bereiche Raum und Form sowie Zahlen und Operationen unterteilt. Innerhalb dieser Kategorien wurde ausgewertet, wie viele Fehler die Kinder durchschnittlich in dem jeweiligen Inhaltsbereich beheben konnten (Abbildung 12), wobei in beiden Kategorien jeweils sechs zu behebende Fehler eingebaut wurden.

Inhaltsbereich	Fehleranzahl in Programmierung	Durchschnittliche Anzahl behobener Fehler
Raum und Form	6	2,83
Zahlen und Operationen	6	1,67

Abbildung 12: Durchschnittliche Anzahl der behobenen Fehler in den Inhaltsbereichen Raum und Form sowie Zahlen und Operationen.

In beiden Inhaltsbereichen konnten weniger als die Hälfte der Fehler behoben werden. Im Inhaltsbereich von Raum und Form konnten drei Kinder alle Fehler beheben, während vier Kinder keine richtigen Lösungen aufwiesen. Jeweils zwei Kinder konnten vier bzw. drei Fehlprogrammierungen korrekt überarbeiten und ein weiteres Kind zwei. In dieser Kategorie konnten die Kinder im Schnitt 2,83 Fehler beheben.

Im Inhaltsbereich der Zahlen und Operationen schnitten die Kinder schlechter ab. Kein Kind konnte die Programmierung vollständig erfolgreich überarbeiten, während zwei Kinder keinen Fehler beseitigen konnten. Die höchste Anzahl an Fehlerbehebungen konnten vier Schüler mit drei erfolgreichen Umsetzungen erbringen. Weitere vier Schüler haben jeweils einen Fehler behoben und die restlichen Schüler konnten zwei Fehler korrigieren. In dieser Kategorie konnten die Kinder im Schnitt 1,67 Fehler beheben.

Bei den Resultaten der Entfernungsknöpfe war auffällig, dass viele Kinder die Teilbarkeit durch sieben im Sinne der Aufgabe richtig gelöst haben, dafür jedoch nicht die Kompetenz innerhalb des Operationsverständnisses genutzt haben. Viele Schüler haben den Wiederholungswert zwei der kurzen Entfernung übernommen und den Schrittewert erhöht. Beispielsweise wurden die Werte 2×21 für die mittlere Distanz und 2×35 für die lange Distanz eingetragen. Da diese Lösung

nicht die angedachte Kompetenz des Zuordnens von Wiederholungen und Multiplikationsaufgaben beinhaltet, sind diese Lösungen nicht als korrekt gewertet worden.

Die Auswertung der Aufgabenstellung zur Überprüfung der dritten Hypothese weist bei der ersten Aufgabe eine hohe Erfolgsquote auf. Sechs Schüler konnten die Anforderungen fehlerfrei in ScratchJr programmieren, während ein Schüler nur ein Element korrekt umsetzen konnte. Jeweils drei Schüler wiesen einen bzw. zwei Fehler in ihren Algorithmen auf.

Die Ergebnisse des Fragebogens zeigen auf, dass die meisten Schüler das Handling und den Lernzuwachs durch die App positiv bewerten, während der Umfang für die meisten Schüler ungenügend ist (Abbildung 13).

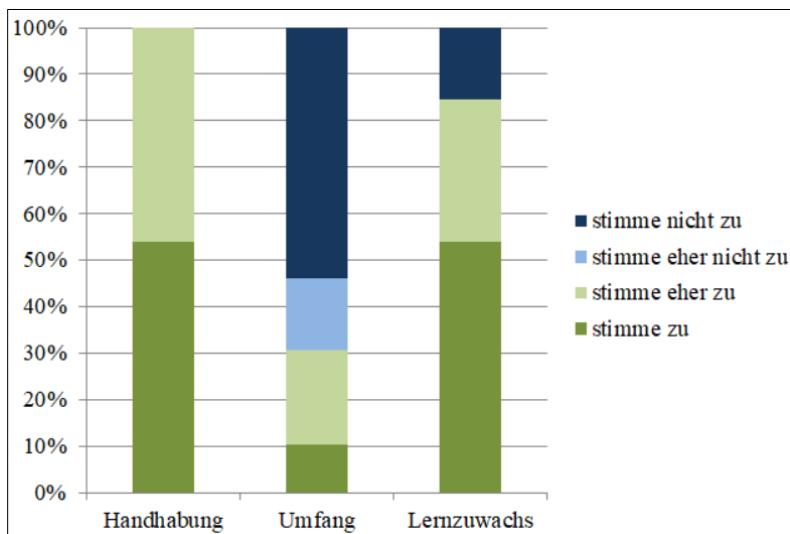


Abbildung 13: Schülerbewertungen von ScratchJr in den Kategorien Handhabung, Umfang und Lernzuwachs.⁶²

Kein Schüler empfindet das Verwenden von ScratchJr als schwierig und alle stimmen zu, dass das Arbeiten mit ScratchJr leicht fällt. Nur knapp über 30% der Schüler empfinden die Entdeckungs- und Auswahlmöglichkeiten sowie das Finden neuer Ideen als ausreichend. Mit etwas über 80% gibt der Großteil der Kinder an, bei der Arbeit mit der App etwas dazugelernt zu haben.

Bei den Angaben zu konkreten Verwendungsmöglichkeiten von ScratchJr haben acht Schüler angegeben Figuren oder Hintergründe zu gestalten, 13 Mal wurde die Antwort Programmieren

⁶² Für eine bessere Übersichtlichkeit der Darstellung wurden die Antwortmöglichkeiten zur ersten Aussage durch die Antwortmöglichkeiten der anderen Aussagen ersetzt, d.h. sehr leicht: stimme zu, leicht: stimme eher zu, schwer: stimme eher nicht zu, sehr schwer: stimme nicht zu.

ausgewählt und drei Schüler gaben Sonstiges an. Letztere Angabe wurde nur einmal präzisiert, indem beim Freitextfeld „Quatsch Videos machen“ aufgeschrieben wurde.

4.4) Analyse

Die Auswertung der Resultate der ersten Hypothese deutet darauf hin, dass der Einstieg in Scratch durch die vorherige Verwendung von ScratchJr erschwert wird. Die Anzahl der erfolgreichen Umsetzung der Vorgaben ist deutlich niedriger im Vergleich zu ScratchJr, bei welcher alle Kinder die Vorgaben umsetzen konnten. Besonders auffällig sind die Ergebnisse der ersten und fünften Vorgabe, welche von keinem Schüler umgesetzt werden konnten. Insbesondere die erste Vorgabe ist in ScratchJr relativ einfach umzusetzen, während Scratch die Angabe von Gradzahlen oder Werten innerhalb eines Koordinatensystems fordert. Diese Diskrepanz könnte als Grund für die niedrige Erfolgsquote angeführt werden. Die Ergebnisse der letzten Vorgabe könnten einerseits auf zeitliche Gründe zurückzuführen sein und andererseits auf einen höheren Anforderungsgrad im Vergleich zu den anderen Vorgaben, da diese Vorgabe durch das Verwenden mehrerer Programmierblöcke umgesetzt werden muss. Die relativ hohe Anzahl erfolgreicher Programmierungen der zweiten und dritten Vorgabe könnte darauf zurückzuführen sein, dass die Blöcke zur Durchführung dieser Vorgaben in beiden Programmen farbliche Ähnlichkeit aufweisen. Diese Annahme steht allerdings entgegengesetzt der niedrigen Anzahl richtiger Programmierungen der vierten Vorgabe, bei denen ebenfalls eine Farbähnlichkeit vorhanden ist. Die niedrigere Erfolgsquote dieser Vorgabe könnte ebenfalls auf zeitliche Probleme zurückzuführen sein, da die Kinder bei der Bearbeitung chronologisch vorgegangen sind und beispielsweise durch Schwierigkeiten beim Umsetzen der ersten Vorgaben für die letzten Vorgaben weniger Zeit vorhanden war.

Die Erschwernis des Einstiegs in Scratch zeigt sich auch bei den Befragungsergebnissen der Schüler, bei denen - bis auf das Einhalten der dritten Vorgabe - über die Hälfte Schwierigkeiten bei der Bearbeitung angaben. Diese Angaben decken sich insbesondere mit den niedrigen Erfolgsquoten der ersten und fünften Vorgabe.

Die angeführten Beschreibungen für Probleme lassen sich in die Problembereiche der fehlenden Zeit und das Finden der Blöcke kategorisieren. Dabei ist es möglich, dass Probleme beim Finden der Blöcke die Zeitprobleme bedingen. Um das Problem des Ermittelns der Blöcke zu konkretisieren wurden die Schüler dazu im Plenum genauer befragt. Dabei wurde vor Allem

Unübersichtlichkeit des Browserprogramms angegeben, was darauf deuten könnte, dass der unterschiedliche Aufbau und die Struktur der Programme für die Kinder problematisch waren.

Diese Erkenntnisse lassen darauf schließen, dass die Annahme der ersten Hypothese bestätigt werden kann.

Die grundsätzlichen strukturellen Probleme der beiden Programmvarianten zeigen Herausforderungen des Übergangs auf. Die Ergebnisse deuten darauf hin, dass die Verwendung von ScratchJr den Einstieg in das Browserprogramm nicht erleichtert, da die strukturellen Unterschiede zu groß sind. Insbesondere die in ScratchJr einfach umzusetzenden Richtungsbewegungen zeigen große Differenzen zum Scratch auf. Die genauen Gründe für die Probleme der Schüler, wie z.B. Unterschiede in Blockfarben, Blockausrichtung etc. müssten durch genauere Analysen erörtert werden. Zudem könnten die Bedingungen des Einstiegs näher beleuchtet werden, da nur ein grober Überblick vor Bearbeitung der Aufgaben gegeben wurde. Für einen gelingenden Übergang von ScratchJr zu Scratch sollten die strukturellen Unterschiede beachtet werden und vor dem Übergang thematisiert werden. Eine ausführlichere Heranführung zum Aufbau von Scratch vor dem Verwenden könnte ebenfalls sinnvoll sein. Dennoch stellt sich die Frage, inwiefern eine Verwendung von ScratchJr als Hinführung zu Scratch in Anbetracht der aufgezeigten Probleme grundsätzlich sinnvoll ist.

Die zweite Hypothese lässt sich durch die gesammelten Ergebnissen nicht bestätigen. In beiden Bereichen konnten im Durchschnitt weniger als die Hälfte der Fehler behoben werden. Daraus könnte geschlossen werden, dass die Arbeit mit ScratchJr die ins Auge gefassten Inhaltsbereiche nicht fördert. Insbesondere im Bereich der Zahlen und Operationen weisen die Schüler eine geringe Erfolgsquote auf. Bei der Auswertung ist zu beachten, dass der Großteil der Kinder das Kopfrechnen mit der Zahl 7 beherrscht, den Zusammenhang von Wiederholungen und Multiplikation in der Aufgabe jedoch nicht durchdrungen haben.

Probleme bei den Aufgabenbearbeitungen könnten zudem durch die Handhabung des Programms entstanden sein. Zwar wurde Scratch in drei Sitzungen vorher erprobt, dennoch könnte diese Zeit nicht ausreichend gewesen sein, damit die Schüler ausreichend vertraut mit der Struktur, Umfang oder Auswahlmöglichkeiten sind. Dadurch könnten kognitive Kapazitäten für außermathematische Prozesse verwendet worden sein.

Zudem ist darauf hinzuweisen, dass die behandelten mathematischen Inhalte nicht explizit beim Verwenden von ScratchJr thematisiert wurden. Die Hypothese begründete sich dadurch, dass die

Inhalte theoretisch in ScratchJr vermittelt werden können und durch die Kinder selbstständig erlernt würden.

Herausforderungen des Übergangs von ScratchJr zu Scratch im Hinblick auf die Behandlung mathematischer Inhalte könnten demnach durch die Art der Vermittlung in den Programmen entstehen. Da keine konkrete Lehreinheit zu den behandelten Inhalten in ScratchJr durchgeführt wurde, könnten die schlechten Ergebnisse auch dadurch entstanden sein. Möglicherweise wären die Aufgaben in Scratch bei Vorausführen der konkreten Inhalte in ScratchJr erfolgreicher umgesetzt worden. Zudem sollte beachtet werden, dass ScratchJr auf Grund des relativ geringen Umfangs nur ausgewählte mathematische Inhalte vermitteln kann. Außerdem hat die Überprüfung keine prozessbezogenen Kompetenzen ins Auge gefasst, was für zukünftige Analysen von Interesse sein könnte, da sich informative und mathematische Prozessbereiche teilweise überschneiden. ScratchJr könnte möglicherweise prozessbezogene Kompetenzen vermitteln, welche in der Auswertung nicht mit einbezogen wurden.

Die Programme enthalten Potenziale für die Vermittlung mathematischer Kompetenzen, Herausforderungen zeigen sich aber in der Vermittlungsart, dem Umfang von ScratchJr und den Unterschieden der Programmierblöcke der Programmvarianten. Diese Aspekte sollten beim Programmübergang beachtet werden.

Die Auswertung der Arbeitsaufträge der letzten Hypothese geben Hinweise für eine Eignung von ScratchJr für die Grundschule. Die erste Aufgabe konnte größtenteils vollständig oder mit wenigen Fehlern umgesetzt werden, weshalb die damit angesprochene Kompetenzerwartung des Darstellens von Algorithmen in verschiedenen Darstellungsformen der GI im Inhaltsbereich der Algorithmen⁶³ von den Schülern durchdrungen sein könnte. Darüber hinaus zeigt die Befragung, dass die Schüler mit der Handhabung der App keine Probleme haben. Da die Schüler über dem Alter der Zielgruppe von ScratchJr liegen, ist diese Einsicht nicht verwunderlich, da ältere Kinder höhere kognitive Leistungen abrufen können und somit den Anforderungen der App gewachsen sein sollten. Des Weiteren scheint es zu einem Lernzuwachs bei den Schülern gekommen zu sein. In welcher Form dieser Lernzuwachs bei den Kindern tatsächlich stattgefunden hat wurde nicht näher betrachtet, könnte für zukünftige Analysen aber von Interesse sein. Kritisiert wurde der Umfang der App, da die Schüler sich beispielsweise mehr Auswahlmöglichkeiten wünschen. Die

⁶³ Vgl. Gesellschaft für Informatik e.V., 2019, S. 13.

Beobachtungen während der AG bestätigten diese Ansicht, da nach wenigen Sitzungen Schüler oft die gleichen Inhalte programmiert oder keine weiteren Ideen für Programmierungen hatten.

Die Angaben darüber, womit sich die Kinder beim Verwenden der App am meisten beschäftigen lässt weitere Schlüsse zu. Größtenteils wurden die Auswahlmöglichkeiten des Programmierens von Geschichten und des Programmierens von Spielen ausgewählt. Dieses Programmieren könnte mit der Aneignung von informatischen Kompetenzen einhergehen, welche exemplarisch in Kapitel 3 aufgezeigt wurden. Das Ankreuzen der Auswahlmöglichkeit des Entdeckens und Ausprobierens neuer Möglichkeiten könnte ebenfalls auf den Erwerb informatischer Kompetenzen hinweisen, da z.B. Modellieren von Problemlösungen gefördert wird. Am zweithäufigsten wurde das Gestalten von Hintergründen oder Figuren angegeben. Zwar könnte hierdurch möglicherweise die Kompetenz der Kreativität gefördert werden, allerdings trägt diese Verwendungsmöglichkeit wahrscheinlich nicht in hohem Maße zur Bildung informatischer Kompetenzen bei. Die einzige konkretisierte Ausführung sonstiger Tätigkeiten wurde mit „Quatsch Videos machen“ beschrieben, was auf einen spielerischen Umgang mit der App hinweist und weniger auf eine Verwendung innerhalb eines klassischen Unterrichts.

Die Einbindung der Ergebnisse der ersten beiden Hypothesen liefern Hinweise für die Bestätigung der dritten Hypothese. Da die Vermittlung mathematischer Kompetenzen sowie die Programmunterschiede Problematiken aufweisen, könnten diese Herausforderungen gegen eine Eignung der App sprechen.

Es ist zu beachten, dass die Auswertungen nicht den ganzen Bereich der Grundschule im Sinne der dritten Hypothese abdecken konnten. Einerseits, da keine Schüler der ersten beiden Klassenstufen involviert waren und andererseits durch die Erprobung der Hypothesen in einer AG. Innerhalb einer AG herrscht insbesondere ein spielerisches Klima, in welchem der Lehrcharakter der Grundschule weniger im Vordergrund steht, weshalb eine nähere Analyse für die Eignung der App im Schulunterricht von Nöten ist. Auf Grund dessen lässt sich die aufgestellte Hypothese nach der Auswertung nicht bestätigen, wobei die Ergebnisse eine präziser formulierte Gegenthese unterstützen könnten: ScratchJr eignet sich für die Vermittlung informatischer Kompetenzen für Schüler der dritten und vierten Klassenstufe im Rahmen einer Arbeitsgemeinschaft.

Da sich keine Bestätigung oder Falsifizierung der Eignung von ScratchJr für die Grundschule ergeben hat, können Herausforderungen des Übergangs schwer ausgemacht werden. Die Handhabung deutet auf eine Eignung hin, während der Umfang (zumindest für die dritte und

vierte Klassenstufe) unzureichend für die Grundschule scheint. Mögliche Schlüsse daraus wären, ScratchJr eher in den ersten beiden Klassenstufen zu verwenden und einen Übergang ab Klasse 3 anzustreben.

5.) Fazit

Die Analyse zeigt auf, dass die Verwendung beider Programmvarianten zu Herausforderungen beim Übergang von ScratchJr zu Scratch führen kann. Diese ergeben sich zum einen durch die Unterschiede der beiden Programme. Dies zeigt sich dadurch, dass das Umsetzen bestimmter Vorgaben in ScratchJr problemlos durchgeführt werden konnte, während große Schwierigkeiten bei der Umsetzung in Scratch zu erkennen waren. Diese Umstände deuten darauf hin, dass die Verwendung von ScratchJr den Übergang zu Scratch erschwert und der Einstieg durch die vereinfachte Programmvariante nicht begünstigt wird. Welche Abweichungen der Programme für die Schwierigkeiten des Übergangs eine wesentliche Rolle spielen konnte nur ansatzweise herausgearbeitet werden. Die Angaben über zeitliche Probleme und das Suchen der richtigen Programmierblöcke könnte auf Herausforderungen mit dem Umfang von Scratch hinweisen, da dieser deutlich höher gegenüber der vereinfachten Variante ist. Vor dem Programmübergang sollten deshalb die Differenzen von ScratchJr und Scratch bewusst gemacht werden und beim Einstieg in Scratch beachtet werden. Falls Scratch als Unterrichtsmittel eingeführt werden soll, deuten die Ergebnisse darauf hin, dass eine Verwendung von ScratchJr als Hinführung zu Scratch weniger sinnvoll ist.

Das Zusammenspiel der Softwarevarianten in Bezug auf die Vermittlung mathematischer Inhalte birgt ebenfalls Herausforderungen. Die Analyse hat aufgezeigt, dass die Bearbeitung mathematischer Inhalte in Scratch nicht durch die vorherige Nutzung von ScratchJr begünstigt wird. Dabei ist zu beachten, dass diese in ScratchJr nicht explizit vermittelt wurden, sondern von einer unterbewussten Förderung mathematischer Kompetenzen ausgegangen wurde. Um eine Vermittlung erfolgreicher zu gestalten besteht die Herausforderung darin, die Inhalte konkret zu thematisieren und anhand der Programme aufzuzeigen. Darüber hinaus ist die Aneignung mathematischer Kompetenzen in ScratchJr durch den Umfang des Programms begrenzt, wohingegen Scratch deutlich mehr Potenziale für das Herbeiführen von Kompetenzen aufzeigt. Diese Begrenzungen sollten beim Einstieg in Scratch bedacht werden, wenn mathematische Inhalte mit den beiden Programmvarianten behandelt werden.

Über die grundsätzliche Eignung von ScratchJr konnten keine eindeutigen Schlüsse gezogen werden. Die Handhabung und Vermittlung spezifischer informatischer Kompetenzen im Bereich der Algorithmen zeigen positive Aspekte der App auf, während Umfang und der Einbezug der anderen Ergebnisse gegen eine Eignung sprechen. Für den Einsatz in der Grundschule lässt sich daraus ableiten, dass die Altersangaben für die beiden Programmvarianten beachtet werden sollten. Schüler der ersten oder zweiten Klassenstufe könnten mit dem Umfang und der Handhabung von ScratchJr weniger Probleme haben und sich an den Bereich der Algorithmen nähern. In höheren Klassenstufen zeigen sich nach einigen Verwendungen die Grenzen der App auf, sodass die Verwendung des Browserprogramms sinnvoller sein könnte. Dabei ist es fraglich, inwiefern die Verwendung von ScratchJr in der dritten und vierten Klasse vor dem Arbeiten mit Scratch sinnvoll ist.

Es zeigen sich verschiedene Herausforderungen beim Übergang von ScratchJr zu Scratch in der Grundschule, welche sich in den angeführten Bereichen widerspiegeln. Es ist zudem zu beachten, dass die Analyse eine kleine Stichprobengröße aufweist und innerhalb einer AG durchgeführt wurde, weshalb nicht alle Bereiche der Grundschule abgedeckt wurden. Differenziertere Analysen könnten weitere relevante Erkenntnisse des Übergangs aufzeigen. Deshalb sollten weiterführenden Beobachtungen durchgeführt werden, da Scratch und ScratchJr Potenziale für die Entwicklung wichtiger Kompetenzen aufzeigen.

6.) Literaturverzeichnis

Calao, Luis Alberto et al. (2015): Developing Mathematical Thinking with Scratch. An Experiment with 6th Grade Students. In: Conole et al.: Design for Teaching a Learning in a Networked World. Heidelberg u.a.: Springer. S. 17-27.

DevTech Research Group der Tufts University et al. (2021): Über ScratchJr. Online unter: <https://www.scratchjr.org/about/info> [Stand 02.12.21].

DevTech Research Group der Tufts University et al. (2021): Aktivitäten. Online unter: <https://www.scratchjr.org/teach/activities> [Stand 16.12.21].

DevTech Research Group der Tufts University et al. (2021): Ressourcensuche. Online unter: <https://scratchjrconnect.tufts.edu/browse/> [Stand 16.12.21].

Google (2021): Google for Education. CS First. Online unter: <https://csfirst.withgoogle.com/s/de/home> [Stand 31.11.21].

Gesellschaft für Informatik e.V. (2019): Kompetenzen für informatische Bildung im Primarbereich. Online unter: <https://dl.gi.de/handle/20.500.12116/20121> [Stand 04.12.21]

Harvard Graduate School of Education (201): From Music to Math: Scratch Across Every Subject. Online unter: <https://scratched.gse.harvard.edu/resources/scratch-across-every-subject-recap.html> [Stand 10.11.21].

Harvard Graduate School of Education (2017): Scratch Across Every Subject: Math. Online unter: <https://scratched.gse.harvard.edu/resources/scratch-every-subject-math.html> [Stand 10.11.21].

Iskrenovic-Momcilovic, Olivera (2020): Improving Geometry Teaching with Scratch. In: International Electronic Journal Of Mathematics Education. Vol. 15, S. 1-8. Online unter: <https://doi.org/10.29333/iejme/7807>.

Kyza, Eleni A. et al. (2021): A Cross-Sectional Study Investigating Primary School Children's Coding Practices and Computational Thinking Using ScratchJr. In: Journal of Educational Computing. S. 1-38. Online unter: <https://doi.org/10.1177/07356331211027387>.

Medienberatung NRW et al. (2021): Medienkompetenzrahmen NRW. Online unter: <https://medienkompetenzrahmen.nrw/medienkompetenzrahmen-nrw/> [Stand 20.12.21].

Medienberatung NRW et al. (2021): Scratch: Einführung in die Blockprogrammierung. Online unter: <https://medienkompetenzrahmen.nrw/unterrichtsmaterialien/detail/scratch-einfuehrung-in-die-blockprogrammierung> [Stand 21.12.21].

Ministerium für Schule und Bildung des Landes Nordrhein-Westfalen (2021): Lehrpläne für die Primarstufe in Nordrhein-Westfalen.

Scratch Foundation (2021): Allgemeine Fragen. Online unter: <https://scratch.mit.edu/faq#scratch3> [Stand 10.11.21].

Scratch Foundation (2021): Curriculum Connections with Scratch. Online unter: <https://sip.scratch.mit.edu/themes/curriculum/> [Stand 10.11.21].

Scratch Foundation (2021): Für Lehrkräfte. Online unter: <https://scratch.mit.edu/educators/> [Stand 10.11.21].

Scratch Foundation (2021): Jahresbericht 2020. Online unter: <https://scratch.mit.edu/annual-report/> [Stand 16.12.21].

Scratch Foundation (2021): Jahresbericht 2020. Online unter: <https://scratch.mit.edu/annual-report/> [Stand 04.12.21].

Scratch Foundation (2021): Our Team. Online unter: <https://scratch.mit.edu/credits> [Stand 10.11.21].

Scratch Foundation (2021): Statistiken. Online unter: <https://scratch.mit.edu/statistics/> [Stand 10.11.21].

Scratch Foundation (2021): Über Scratch. Online unter: <https://scratch.mit.edu/about> [Stand 20.12.21].

Shin & Park (2014): A Study on the Effect affecting Problem Solving Ability of Primary Students through the Scratch Programming. In: Advanced Science and Technology Letters. Vol. 59, S. 117-120. Online unter: <http://dx.doi.org/10.14257/astl.2014.59.27>.

7.) Abbildungsverzeichnis

Abbildung 1: Vier P's kreativen Lernens	3
Abbildung 2: Altersverteilung neuer Scratch-Nutzer.....	4
Abbildung 3: Screenshot des Browserprogramms Scratch mit einer Beispielprogrammierung.	6
Abbildung 4: Blockformen illustriert anhand von Beispielen und dessen Funktionen.	8
Abbildung 5: Screenshot der App ScratchJr mit einer Beispielprogrammierung.	10
Abbildung 6: Gitteransicht in ScratchJr.	11
Abbildung 7: Vergleich der Programmvarianten Scratch und ScratchJr.	12
Abbildung 8: Von den Schülern zu vervollständigende Programmierung (links) und eine mögliche Lösung des Arbeitsauftrags (rechts).	21
Abbildung 9: Scheibengrafik zum Bestimmen der Drehrichtung.	22
Abbildung 10: Links: Ansicht der Programmierung mit den geometrischen Figuren Rechteck, Quadrat, Dreieck sowie den ausführenden Knöpfen. Rechts: Ausschnitt eines zu vervollständigenden Algorithmus für das Quadrat (oben) und eine mögliche Lösung (unten).	23
Abbildung 11: Links: Anzahl richtiger Programmierungen (grün) und fehlerhafter Programmierungen (rot) je Vorgabe. Rechts: Empfinden des Schwierigkeitsgrades in Prozent je Vorgabe von sehr leicht (dunkelgrün), leicht (hellgrün), schwer (hellblau) bis sehr schwer (dunkelblau).	27
Abbildung 12: Durchschnittliche Anzahl der behobenen Fehler in den Inhaltenbereichen Raum und Form sowie Zahlen und Operationen.....	28
Abbildung 13: Schülerbewertungen von ScratchJr in den Kategorien Handhabung, Umfang und Lernzuwachs.	29

8.) Anhang

Anhang 1: Arbeitsauftrag und Fragebogen zur Überprüfung der ersten Hypothese

Arbeitsblatt I

Aufgabe 1)

Erstelle einen Algorithmus bei Scratch. Folgende Vorgaben sollen dabei eingehalten werden:

- 1.) Eine Figur bewegt sich nach links, rechts, oben und unten.
- 2.) Eine Figur vergrößert oder verkleinert sich.
- 3.) Eine Figur sagt etwas.
- 4.) Eine Anweisung wird wiederholt.
- 5.) Eine Figur verändert ihr Verhalten, sobald diese berührt wird.

Fragebogen:

Wie leicht fiel dir die Bearbeitung der Aufgaben? Kreuze an.

Das Bearbeiten von Vorgabe 1 fiel mir

sehr leicht leicht schwer sehr schwer

Falls du Probleme hattest oder die Vorgabe nicht lösen konntest, beschreibe woran das gelegen hat:

Das Bearbeiten von Vorgabe 2 fiel mir

sehr leicht leicht schwer sehr schwer

Falls du Probleme hattest oder die Aufgabe nicht lösen konntest, beschreibe woran das gelegen hat:

Das Bearbeiten von Vorgabe 3 fiel mir

sehr leicht leicht schwer sehr schwer

Falls du Probleme hattest oder die Aufgabe nicht lösen konntest, beschreibe woran das gelegen hat:

Das Bearbeiten von Vorgabe 4 fiel mir

sehr leicht

leicht

schwer

sehr schwer

Falls du Probleme hattest oder die Aufgabe nicht lösen konntest, beschreibe woran das gelegen hat:

Das Bearbeiten von Vorgabe 5 fiel mir

sehr leicht

leicht

schwer

sehr schwer

Falls du Probleme hattest oder die Aufgabe nicht lösen konntest, beschreibe woran das gelegen hat:

Anhang 2: Arbeitsaufträge zur Überprüfung der zweiten Hypothese

Arbeitsblatt II

Auf dem Bild kannst du die Katze Scratchy im Stadion sehen. Scratchy möchte in die vier Richtungen links, rechts, oben und unten gehen. Außerdem möchte er eine kurze, mittlere und lange Entfernung gehen. Kannst du Scratchy helfen, dass er beim Drücken der verschiedenen Knöpfe die richtigen Anweisungen ausführt?

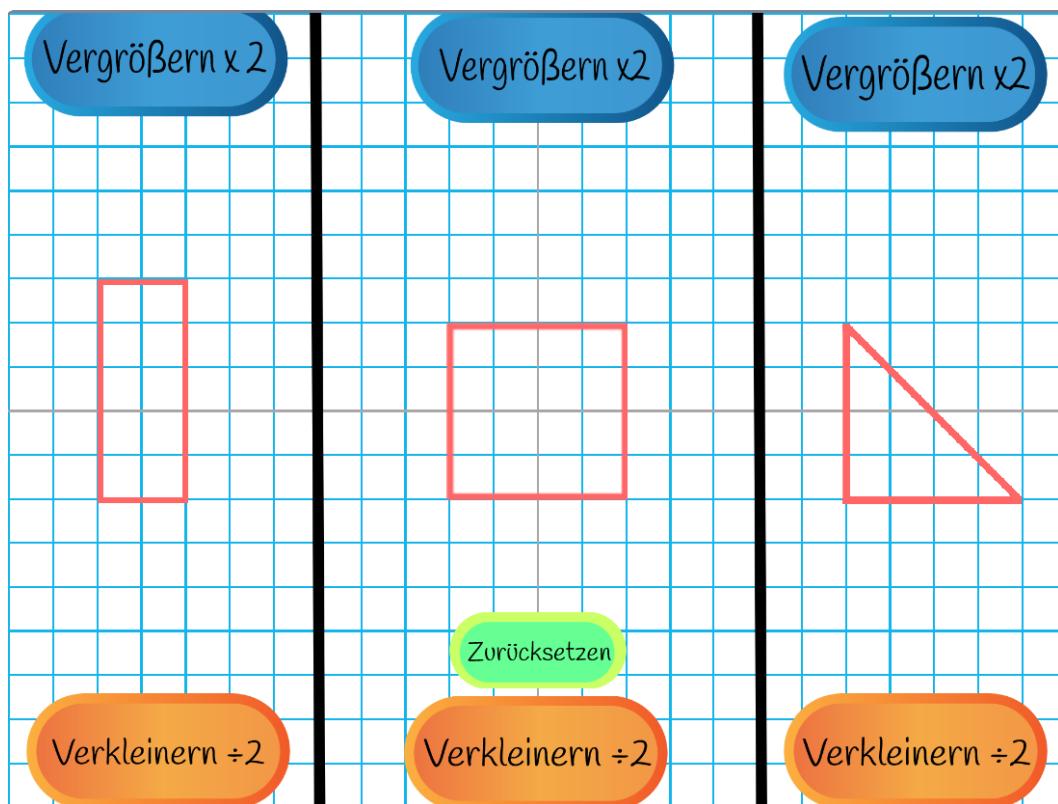


Aufgabe 1)

Führe die Programmierung in Scratch so zu Ende, dass alle Knöpfe richtig funktionieren.

Die Schritte die Scratchy bei der kurzen, mittleren und langen Entfernung geht, sollen dabei **immer durch 7 teilbar** sein.

Im nachfolgenden Bild kannst du drei Figuren erkennen. Diese sollen sich per Knopfdruck vergrößern und verkleinern.



Aufgabe 2)

Führe die Programmierung in Scratch so zu Ende, dass alle Knöpfe richtig funktionieren.

Die Figuren sollen dabei beim Vergrößern **ungefähr doppelt so groß** werden und beim Verkleinern **ungefähr doppelt so klein**.

Du kannst die Figuren auf die Normalgröße zurücksetzen, indem du auf den grünen Knopf drückst.

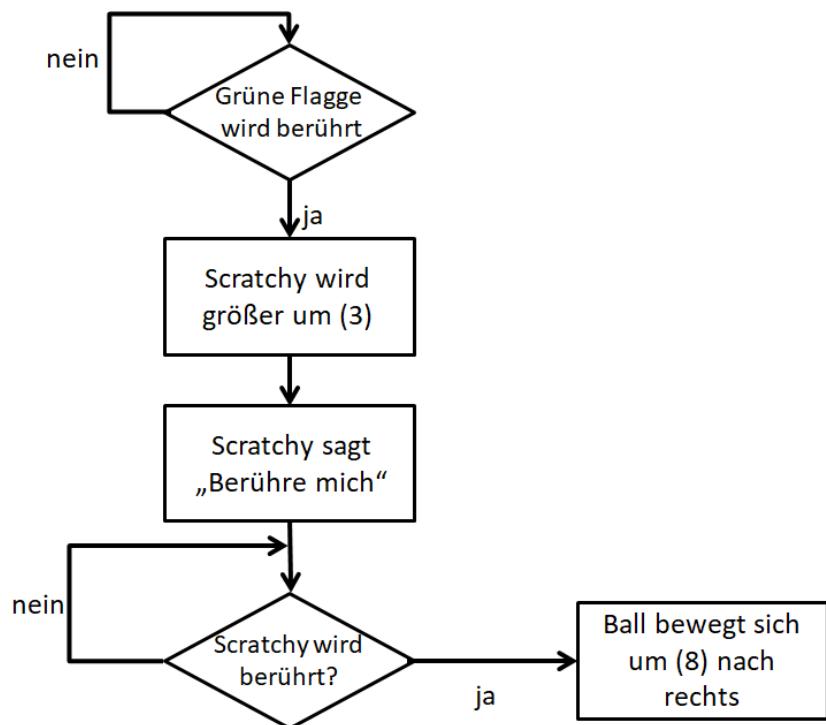
Anhang 3: Arbeitsauftrag und Fragebogen zur Überprüfung der dritten Hypothese

Arbeitsblatt III

Aufgabe 1)

Auf deinem I-Pad findest du das Projekt „Aufgabe 2“.

Übertrage das nachfolgende Bild in ScratchJr, damit die Figuren der Katze (Scratchy) und des Fußballs den Algorithmus ausführen.



Fragebogen zu ScratchJr

Kreuze an, welche Aussagen deiner Meinung nach zutreffen.

1.) Mir fällt das Arbeiten mit ScratchJr

sehr leicht leicht schwer sehr schwer

2.) Ich habe das Gefühl, dass ich noch Vieles in Scratch Jr entdecken kann.

stimme zu stimme eher zu stimme eher nicht zu stimme nicht zu

3.) Mir fallen keine neuen Ideen ein, die ich mit ScratchJr durchführen kann.

stimme zu stimme eher zu stimme eher nicht zu stimme nicht zu

4.) Ich wünschte mir, dass ich mehr Auswahlmöglichkeiten bei ScratchJr hätte.

stimme zu stimme eher zu stimme eher nicht zu stimme nicht zu

5.) Seitdem ich mit ScratchJr arbeite, habe ich Vieles dazu gelernt.

stimme zu stimme eher zu stimme eher nicht zu stimme nicht zu

6.) Die meiste Zeit verbringe ich bei ScratchJr mit (nur **maximal zwei** Kreuze machen!)

- Gestalten von Figuren oder Hintergründen
 - Programmieren von Geschichten
 - Programmieren von Spielen
 - Entdecken/ Ausprobieren neuer Möglichkeiten
 - Sonstiges
-
-
-
-
-
-
-
-
-

Anhang 4: Gegenüberstellung der Blöcke in Scratch und ScratchJr.

Kategorie	Funktion	Programmvariante	
		Scratch	ScratchJr
Bewegung	Kontrollieren die Bewegungen von Figuren	<p>gehe 10 er Schritt drehe dich ⌂ um 15 Grad</p>	<p>1 1</p>
Aussehen	Kontrollieren das Aussehen von Figuren oder Hintergründen	<p>sage Hello! ändere Größe um 10</p>	<p>2 hallo</p>
Klang	Kontrollieren die Klänge von Figuren	<p>spielt Klang Mau ▾ ganz</p>	<p>pop</p>
Ereignisse	Starten Skripte oder senden Nachrichten	<p>Wenn diese Figur angeklickt wird sende Nachricht1 ▾ an alle</p>	
Steuerung	Steuern Skripte	<p>wiederhole 10 mal</p>	<p>4</p>
Fühlen	Führen mathematische Funktionen und Textverarbeitung aus	<p>wird Rand ▾ berührt?</p>	
Operatoren	Führen mathematische Funktionen und Textverarbeitung aus	<p>Zufallszahl von 1 bis 10</p>	
Variablen	Verwalten Zahlen oder Zeichen	<p>ändere meine Variable ▾ um 1</p>	
Meine Blöcke	Beinhalten selbsterstellte Blöcke		
Abschluss	Stellen das Ende eines Skripts fest		

Anhang 5: Ausgefüllte Fragebögen zur Überprüfung der ersten Hypothese

- FragebögenAB1_ausgefüllt.pdf (siehe CD)

Anhang 6: Ausgefüllte Fragbögen zur Überprüfung der dritten Hypothese

- FragebögenAB3_ausgefüllt.pdf (siehe CD)

Anhang 7: Schülerprogrammierungen des Arbeitsblatts I

- Dateien in Ordner *Programmierungen AB1* (siehe CD)

Anhang 8: Schülerprogrammierungen des Arbeitsblatts II

- Dateien in Ordner *Programmierungen AB2* (siehe CD)

Anhang 9: Schülerprogrammierungen des Arbeitsblatts III

- Dateien in Ordner *Programmierungen AB3* (siehe CD)

Hinweis: Die Programmierungen der Schüler Scratch können geöffnet werden, indem diese in Scratch über den Reiter *Datei* hochgeladen werden. Um die Programmierungen der Schüler in ScratchJr öffnen zu können, muss ScratchJr auf einem mobilen Endgerät installiert sein. Die Dateien können dann ausgewählt und mit ScratchJr geöffnet werden.

Anhang 10: Auswertung Ergebnisse zur Überprüfung der ersten Hypothese

- Auswertung Ergebnisse Arbeitsauftrag I.xlsx (siehe CD)

Anhang 11: Auswertung Ergebnisse zur Überprüfung der zweiten Hypothese

- Auswertung Ergebnisse Arbeitsauftrag II.xlsx (siehe CD)

Anhang 12: Auswertung Ergebnisse zur Überprüfung der dritten Hypothese

- Auswertung Ergebnisse Arbeitsauftrag III.xlsx (siehe CD)

Plagiatserklärung der / des Studierenden

Hiermit versichere ich, dass die vorliegende Arbeit über
Herausforderungen in der Grundschule beim Übergang von Scratch zu ScratchJr
selbstständig verfasst worden ist, dass keine anderen Quellen und Hilfsmittel als die angegebenen
benutzt worden sind und dass die Stellen der Arbeit, die anderen Werken – auch elektronischen
Medien – dem Wortlaut oder Sinn nach entnommen wurden, auf jeden Fall unter Angabe der
Quelle als Entlehnung kenntlich gemacht worden sind.

(Datum, Unterschrift)

Ich erkläre mich mit einem Abgleich der Arbeit mit anderen Texten zwecks Auffindung von
Übereinstimmungen sowie mit einer zu diesem Zweck vorzunehmenden Speicherung der Arbeit
in eine Datenbank einverstanden.

(Datum, Unterschrift)