



DIPLOMARBEIT / DIPLOMA THESIS

Titel der Diplomarbeit / Title of the Diploma Thesis

Informatik-Unterricht mittels personenzentrierter Kommunikation zur Vermittlung wesentlicher informatischer und sozialer Kompetenzen

*Ein „In-ClassFlip“-Konzept zur Einführung in die Programmierung
mit App-Inventor*

verfasst von / submitted by

Birgit Pindhofer

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of

Magistra der Naturwissenschaften (Mag.rer.nat.)

Studienkennzahl lt. Studienblatt /
degree programme code as it appears on
the student record sheet:

A 190 406 884

Studienrichtung lt. Studienblatt /
degree programme as it appears on
the student record sheet:

Lehramtsstudium Mathematik & Informatik und
Informationsmanagement

Betreut von / Supervisor:

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

„Ich erkläre hiermit, diese Arbeit selbständig verfasst und die verwendeten Quellen und Hilfsmittel vollständig angegeben zu haben. Jene Stellen der Arbeit – einschließlich der Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen wurden habe ich auf jeden Fall unter Angabe der Quelle als Entlehnung gekennzeichnet.

Ich habe mich weiters bemüht, sämtliche Inhaber der Bildrechte zu identifizieren und ihre Zustimmung zur Verwendung der Bilder in dieser Arbeit eingeholt. Sollte dennoch eine Urheberrechtsverletzung bekannt werden, ersuche ich um Meldung bei mir“.

„Diese Arbeit wurde bisher weder in gleicher noch in ähnlicher Form einer anderen Prüfungsbehörde vorgelegt und sie wurde auch nicht veröffentlicht“.

Enzenreith, Mai 2018

Birgit Pindhofer

Danksagung

Ich danke in erster Linie Frau ao. Univ.-Prof. Dipl.-Ing. Dr. Renate Motschnig, die mich mit ihren Seminaren an der Universität Wien zu dieser Arbeit inspirierte und wesentlich zum Gelingen dieser beitrug, da sie jederzeit für offenen Fragen zur Verfügung stand und mich bei der Suche nach Experten tatkräftig unterstützte.

Speziell bedanken möchte ich mich auch bei jenen Lehrerinnen und Lehrern, welche sich bereit erklärten, das Unterrichtskonzept dieser Arbeit zu begutachten und den Fragebogen zu bearbeiten.

Besonderer Dank geht an meine Studienkolleginnen Christina Fel und Angelika Schwarz, welche mich während des gesamten Studiums motivierten und unterstützten. Sie haben mein Leben bereichert und sind mir in dieser Zeit sehr ans Herz gewachsen.

Von ganzem Herzen bedanke ich mich bei meiner Familie. Im Speziellen gaben mir mein Mann und meine Kinder Raum und Zeit um dieses Studium zu absolvieren und standen mir auch in den schwierigen Phasen beim Schreiben dieser Arbeit bei.

Inhaltsverzeichnis

Einleitung.....	1
Motivation.....	2
Ziele.....	3
Methode	3
Aufbau der Arbeit.....	5
Theoretischer Teil.....	6
Relevanz der Kommunikationsfähigkeit und anderer Soft-Skills für die Informatik	7
Allgemein	7
Kommunikation im Bezug zum Lehrplan Informatik.....	10
Der personenzentrierte Ansatz nach Carl Rogers.....	12
Grundlagen.....	12
Basisvariablen für personenzentrierte Kommunikation	13
Aktives Zuhören	16
Das Humanistische Menschenbild	19
Humanistische Pädagogik - Der personenzentrierte Unterricht.....	20
Signifikantes Lernen.....	24
Die Funktion der Lehrenden.....	26
Die Funktion der Lernenden	29
Umsetzung mittels „In-ClassFlip“	30
Bezug zu den Lerntheorien	39
Visuelle Programmierung	46
Paradigmen der visuellen Programmierung	46
Blockbasierte Programmierumgebungen.....	47
Gründe blockbasierte Programmierumgebungen im Unterricht einzusetzen	48
Computational Thinking.....	52

Praktischer Teil	55
App Inventor	56
Entstehungsgeschichte	56
Besondere Features des App Inventors.....	57
Apps die mit dem App Inventor erstellt werden können	59
Erscheinungsbild des App Inventors	59
Unterrichtskonzept zu Einführung in die Programmierung	61
Bezug des Konzeptes zum Lehrplan.....	61
Evaluierung des Konzepts	64
Auswahl der Experten	64
Konzeption des Fragebogens	65
Erhebungsinstrument	66
Rücklauf	67
Auswertung der Expertenbefragung	68
Conclusio und Ausblick	76
Zusammenfassung	78
Abstract.....	79
Abbildungsverzeichnis	80
Literaturverzeichnis.....	81
Anhang	89
Unterrichtskonzept	
Fragebogen	

EINLEITUNG

MOTIVATION

Während meines Studiums hatte ich im Seminar „Kommunikation und Teamarbeit für Informatiker“ erstmalig das Vergnügen mit der Theorie von Carl R. Rogers in Berührung zu kommen und mich im Rahmen des zu verfassenden e-Portfolios mit dem Buch: „Konstruktive Kommunikation - sich und andere verstehen durch personenzentrierte Interaktion“ (R. Motschnig & L. Nykl, Klett-Cotta, 2009) eingehend auseinanderzusetzen.

Dadurch wurde bei mir das Interesse am personenzentrierten Ansatz nach Carl Rogers und didaktischen Konzepten, welche signifikantes Lernen begünstigen können, geweckt. In meiner zukünftigen Funktion als Lehrerin möchte ich Schülerinnen und Schüler bei der Erreichung ihrer Ziele unterstützen, sie auf ihrem Weg und hin zu selbstständigem, eigenverantwortlichem und lebenslangem Lernen ein Stück begleiten.

Als Mutter von drei Kindern konnte ich hautnah miterleben, wie der angeborene Wissensdrang im Laufe der Schulkarriere deutlich abnahm, weshalb mich der personenzentrierte Ansatz nach Carl R. Roger und die Ideen des Flipped Classrooms sehr ansprechen.

Außerdem würde ich mit meiner Arbeit gerne einen Beitrag für Kolleginnen und Kollegen leisten, indem ich Ideen, Gedanken, und ein Konzept zur Umsetzung von personenzentriertem Unterricht ausarbeite, sowie diese motivieren, gemäß den Theorien von Carl Rogers, ihre Funktion im Klassenzimmer vom Experten und Fachlehrer hin zum Lerndesigner, Lernbegleiter, Mitlernenden und Unterstützer zu verändern.

Im Fokus dieser Arbeit steht die Art und Weise wie signifikantes Lernen bei Schülerinnen und Schülern initiiert werden kann und gleichzeitig Kommunikationsfähigkeiten trainiert werden können.

Der Informatikunterricht eignet sich meines Erachtens hervorragend zur Umsetzung der Konzepte von Carl Rogers, da der Lehrplan „[...] *den Lehrerinnen und Lehrern Freiräume für die eigenständige und verantwortliche Unterrichtsgestaltung* [...]“ bietet. („RIS - Lehrpläne – allgemeinbildende höhere Schulen - Bundesrecht konsolidiert, Fassung vom 18.03.2018). Die Lehrperson hat mit eher kleinen Gruppengrößen zu tun, was Team- und Gruppenarbeit erleichtert und da es oft mehrere Lösungen für ein Problem gibt, entstehen viele Gelegenheiten für Diskussionen.

Da die Informatikbranche unter einem Arbeitskräftemangel leidet und zahlreiche Studien belegen, dass Soft-Skills wie Kommunikation in der IKT sehr gefragt sind, wird es in Zukunft wichtiger werden Jugendliche für Themen wie Programmierung und Projektmanagement zu begeistern, sowie ihre sozialen Fähigkeiten verstärkt auszubilden.

Die Entscheidung App-Inventor zur Einführung in die Programmierung einzusetzen, begründet in der Tatsache, dass hier eine visuelle Programmiersprache, ähnlich wie Scratch verwendet wird, welche es

den Schülerinnen und Schülern ermöglicht, rasch Erfolge zu erzielen, da kein Vorwissen bezüglich Syntax und Semantik von Nöten ist. Auch ist es gewiss motivierend, dass man gemeinsam ein Produkt für einen Gebrauchsgegenstand des täglichen Lebens, das Smartphone, schaffen kann.

Ziele

Das erwartete Ergebnis dieser Diplomarbeit ist ein Konzept für Lehrpersonen, das den personenzentrierten Ansatz repräsentiert und signifikantes Lernen, unter Berücksichtigung des Lehrplans für Informatik in der AHS-Oberstufe, initiiert und bei den Schülern und Schülerinnen ein Interesse an Programmierung und Projektmanagement weckt. Die erstellten Unterrichtsszenarien bieten einen groben Rahmen, der durch Lehrpersonen an die jeweiligen Gegebenheiten im Unterricht anzupassen ist.

Zusätzlich wird versucht, die Kommunikations- und Teamfähigkeit der Schülerinnen und Schülern, sowie deren soziale Kompetenzen zu verbessern.

Lehrende sollen eine Unterstützung bei der Gestaltung schülerzentrierten Unterrichts und einen raschen Überblick über App-Inventor und die theoretischen Hintergründe erhalten, sowie motiviert werden ihre Funktion im Klassenzimmer zu überdenken.

Die Evaluation des Konzepts durch erfahrene Lehrpersonen soll dazu beitragen, dieses zu verbessern und Antworten auf folgende Fragen zu erhalten:

- *Ist ein derartiges Konzept geeignet, um sowohl die vom Lehrplan vorgegebenen informatischen, als auch sozialen Kompetenzen zu vermitteln und bei den Schülerinnen und Schülern Interesse an Programmierung und Projektmanagement zu wecken?*
- *Können sich Lehrende vorstellen die Funktion des Lernbegleiters einzunehmen?*
- *Sehen Lehrpersonen den sozialen Mehrwert den dieses Konzept zu vermitteln versucht?*

Methode

Zu Beginn wird eine ausführliche Literaturrecherche zu den für das Unterrichtskonzept relevanten Theorien durchgeführt. Aufgrund der sich daraus ergebenden Erkenntnisse wird das Konzept erstellt.

Zur Beantwortung der Forschungsfrage und der sich daraus ergebenden Unterfragen wird die Expertenbefragung, eine quantitative Methode aus der Sozialforschung, verwendet.

Bei quantitativen Methoden wird die Beobachtungsrealität nicht in Zahlen, sondern mittels nichtnumerischen Materials abgebildet und interpretativ verarbeitet. Dazu zählen Beobachtungsprotokolle, Interviews, diverse Texte aber auch Filme und Artefakte. (vgl. Bortz, Jürgen & Döring, Nicola, 2007, p. 297)

Da es mir leider im Rahmen dieser Diplomarbeit nicht möglich ist, das erstellte Unterrichtskonzept zu testen und dabei eigene Beobachtungen anzustellen, wähle ich die Methode der Expertenbefragung mittels Fragebogen, um Antworten auf die gestellten Fragen sowie Verbesserungsmöglichkeiten für das Konzept zu erhalten. Hierbei handelt es sich teilweise um eine Evaluationsforschung da Aussagen der Experten über das Funktionieren des untersuchten Konzepts erwartet werden.

„Evaluationsforschung beinhaltet die systematische Anwendung empirischer Forschungsmethoden zur Bewertung des Konzeptes, des Untersuchungsplanes, der Implementierung und der Wirksamkeit sozialer Interventionsprogramme“ (Bortz, Jürgen & Döring, Nicola, 2007, p. 96).

Diese Befragung erfolgt in schriftlicher Form und ist teilstrukturiert, da sie sowohl unstrukturierte Fragen in Form von offenen Fragen, als auch strukturierte Fragen mit genauen Antwortmöglichkeiten beinhaltet.

Diese Form wurde gewählt, um einerseits mit dem geschlossenen Frageformat mehr Objektivität und eine vereinfachte, übersichtlichere Auswertungsmöglichkeit sowie eine höhere Vergleichbarkeit zu haben und andererseits (vgl. Bortz, Jürgen & Döring, Nicola, 2007, p. 254) auch Meinungen und Einstellungen erfassen zu können. Dabei wurde toleriert, dass bei offenen Fragen möglicherweise eher die Antwort verweigert wird. (vgl. Bortz, Jürgen & Döring, Nicola, 2007, p. 401 ff.)

Die gesammelten Ergebnisse der geschlossenen Fragen werden sorgfältig ausgewertet und die Beiträge der offenen Fragen werden einer qualitativen Inhaltsanalyse unterzogen. (vgl. Mayring; Philipp, 2010, p. 601 ff.)

Folgende Funktionen der Forschung werden erwartet:

- **Erkenntnisfunktion:** Die Sammlung von Daten und Erkenntnissen legitimiert oder begründet Entscheidungen.
- **Kontrollfunktion:** Mängel sollten aufgezeigt und korrigiert werden
- **Dialog-/Lernfunktion:** Kontakt zu Personen aus der Zielgruppe des Konzepts und Erkenntnisse aus deren Rückmeldung
- **Legitimierungsfunktion:** Die Frage nach der Sinnhaftigkeit des Einsatzes in einer realen Unterrichtssituation soll beantwortet werden. (Flick, Uwe, 2006, p. 14)

Aufbau der Arbeit

Zunächst wird die Relevanz von „Soft-Skills“ wie Kommunikation und anderen sozialen Fähigkeiten in der Informatik anhand von Studien erläutert, als auch der Zusammenhang des erstellten Konzepts zum Lehrplan der Informatik hergestellt.

Im theoretischen Teil wird eine Recherche und Analyse der Literatur zum personenzentrierten Ansatz, zur personenzentrierten Kommunikation und zum Aktiven Zuhören durchgeführt.

Im weiteren Verlauf wird die humanistische Lerntheorie, der personenzentrierte Unterricht, signifikantes Lernen und das Konzept „In-ClassFlip“, sowie deren Bezug zu lerntheoretischen Ansätzen thematisiert.

Anschließend werden die Vor- und Nachteile visueller Programmierung und im Speziellen des App Inventors beleuchtet, sowie das Thema Computational Thinking bearbeitet.

.

Im praktischen Teil wird ein Unterrichtsmodell zu „Einführung in die Programmierung mittels App-Inventor“ mit Bezug zum Lehrplan für Informatik in der AHS-Oberstufe vorgestellt. Dieses erarbeitete Konzept wird von erfahrenen Lehrenden evaluiert. Das daraus resultierende Feedback wird zur Verbesserung des Konzepts herangezogen, im Folgenden analysiert und liefert Antworten auf die Fragestellung.

THEORETISCHER TEIL

Relevanz der Kommunikationsfähigkeit und anderer Soft-Skills für die Informatik

Es erscheint notwendig neben den technischen Fähigkeiten, Soft-Skills gut auszubilden um am Arbeitsmarkt der Zukunft bestehen zu können, denn in Anforderungsprofilen für Positionen in der Informatikbranche, insbesondere für das Projektmanagement, findet man vermehrt den Begriff Soft-Skills. Weshalb in weiterer Folge erörtert wird, um welche Voraussetzungen es sich hierbei handelt und warum diese für die erfolgreiche Abwicklung von Projekten von großer Bedeutung sind.

Einen wichtigen Beitrag zu erfolgreichen Projekten scheint vor allem die Kommunikationsfähigkeit der Beteiligten zu leisten. Es werden im Folgenden die Rolle der Kommunikation und deren Inhalte in Projekten sowie die Teilhabenden eruiert.

Allgemein

Immer mehr Stellenausschreibungen für Projektleiter erwähnen in den Anforderungen eine breite Palette an Soft-Skills.

Auch eine Studie aus Neuseeland, bei der Stellenanzeigen für die IT-Branche untersucht wurden ergab, dass nur 60 % der 543 analysierten Jobangebote technische Fertigkeiten forderten. Hingegen wurde bei der Analyse und nachfolgenden Interviews eine große Anzahl an „weichen“ Fähigkeiten identifiziert. Rund 300 davon wurden detailliert beschrieben und kategorisiert. Dabei wurden sie zu acht Hauptthemen zusammengefasst: (Stevens Matt & Norman Richard, 2016)

1. Kommunikation:

Schrift-, Sprach- und Körpersprache. Das Vermögen sich klar zu artikulieren, Dinge auf den Punkt zu bringen und Zuhörer zu begeistern, sowie die Anerkennung, dass unterschiedliche Medien unterschiedliche Behandlung benötigen.

2. zwischenmenschliche Fähigkeiten

Das Können Vertrauen und Beziehungen aufzubauen, sich an Namen zu erinnern, soziale Signale aufzugreifen, Gespräche an Kollegen, Vorgesetzten oder Kunden anzupassen, Respekt für andere zu haben und authentisch zu sein.

3. Selbstmanagement

Eigenmotiviert zu sein, Aufgaben abzuschließen, Initiative zu zeigen, gute Arbeitsethik und Sinn von Aktualität zu haben, Aufträge auszuführen und die eigene Meinung einzubringen, sowie die Bereitschaft sich weiterzubilden.

4. Einstellung

Optimistisch, pro-aktiv, positiv, selbstbewusst und neugierig sowie kritikfähig und lernbereit zu sein.

5. Eignung

Die Fähigkeit, vorhandene technische Fertigkeiten anzuwenden bzw. nach Alternativen zu suchen. Anforderungen zu interpretieren und Probleme zu lösen, sowie schnell neues zu lernen.

6. Geschäftsführung verstehen.

Die Qualifikation ergebnis- und kundenorientiert zu sein, ein Kostenbewusstsein zu haben und mit Selbstvertrauen hinter dem Unternehmen zu stehen.

7. Teamwork

Das Geschick Verantwortung für das Team zu übernehmen und andere für ihre Leistung zu respektieren, Konflikte zu lösen und zum Erfolg beizutragen ohne die Beziehungen zu verschlechtern.

8. Respekt

Die Fähigkeit zu erkennen, dass jede Position im Unternehmen wichtig ist und alle ihren Beitrag zu leisten haben, niemanden ausnutzen und selbst jederzeit einsatzbereit sein soll.

In einer Untersuchung von Tiago Miranda & Bimal Ghimire aus dem Jahr 2007 wurde eine Literaturrecherche über Soft Skills durchgeführt und eine Liste von zehn Kategorien der wichtigsten Befähigungen erstellt.

Kommunikation, Führung, Problemlösung, Teamarbeit, Organisation, Flexibilität und Aufmerksamkeit, Kreativität und Innovation, Personalmanagement, Verhandlungs- und Konfliktmanagement und positive Einstellung.

Auch in dieser Untersuchung, wie auch in zahlreichen Artikeln zu diesem Thema, findet man Kommunikation ganz an der Spitze der benötigten Soft-Skills. (Tiago Miranda & Bimal Ghimire, 2007)

Kommunikation scheint also eine der fundamentalsten Fähigkeiten zu sein.

Sie spielt in vielen Bereichen der Informatik eine wesentliche Rolle, angefangen bei der Anforderungsanalyse bis hin zur Teamführung und der Berichterstattung an den Auftraggeber. Projekte müssen präsentiert und „verkauft“, Partner überzeugt, Mitarbeiter motiviert, Verhandlungen geführt, Informationen ausgetauscht, Ergebnisse gesichert, Probleme und Konflikte identifiziert und gelöst, Mitarbeiter geführt werden.

Mathias Freitag hat in seinem Buch „Kommunikation im Projektmanagement“ insgesamt 34 Funktionen der Kommunikation in Projekten aufgelistet und betont, dass agiles Projektmanagement und der Trend zu mehr virtueller Zusammenarbeit den Einfluss der Kommunikation noch steigern. (vgl. Freitag, 2016, p. 257 ff.)

Es nicht nur erforderlich Interaktionswege zu kennen und zu nutzen, mit Gesprächsarten vertraut zu sein (Delegationsgespräch, Anforderungsgespräch, Anerkennungsgespräch, Kriteriengespräch) Feedback zu beherrschen, Gesprächs- und Fragetechniken zu kennen, sondern auch aktiv zuzuhören, um den Gesprächspartnern zu vermitteln, dass sie verstanden und ernstgenommen werden. (vgl. Geipel, 2003, p. 40 ff.)

Das Ziel ist die Vermeidung von Störungen und Missverständnissen, sowie die Verbesserung von Arbeitsabläufen. (vgl. Brei Jennifer & Dr. Wagner Eike)

In jedem Informatikprojekt ist es daher für die Leitung unabdinglich, die Kommunikationsbeziehung zu den diversen Stakeholdern zu bestimmen.

Denn, wie schon in den Axiomen von Paul Watzlawick festgehalten wurde, kann nicht, nicht kommuniziert werden und jede Kommunikation hat einen Inhalts- und Beziehungsaspekt. (vgl. „Paul Watzlawick,“ 2017)

Auf das zweite Axiom von Paul Watzlawick bezieht sich auch das Kommunikationsmodell von Friedemann Schulz von Thun. Er beschreibt mit dem „*Vier-Seiten-Modell*“ dass jede Nachricht vier Aspekte oder Ebenen hat. Diese sind der „*Sachinhalt*“, die „*Selbstoffenbarung*“, die „*Beziehung*“ und der „*Appell*“. Sie werden auch als „*vier Seiten einer Nachricht*“ bezeichnet. Dieses Modell veranschaulicht, welche Missverständnisse zu Störungen in der Kommunikation führen können. (vgl. „Vier-Seiten-Modell,“ 2018)

Obwohl viele Unternehmer, wie man an oben erwähnten Stellenangeboten erkennen kann, bereits die Wichtigkeit der Soft-Skills für eine Position im Projektmanagement erkennen, hinkt die Ausbildung an den Schulen und Universitäten laut Ira Pant, Bassam Baroud noch etwas hinterher. Sie betonen die Notwendigkeit zukünftig mehr Augenmerk auf Soft-Skills bei der Ausbildung zu legen. In ihrer Studie stellen sie fest, dass in der Ausbildung an den australischen Universitäten, aber auch weltweit hauptsächlich nach dem PMI PMBOK („A Guide to the Project Management Body of Knowledge“)

unterrichtet wird. Der Schwerpunkt dieses Werkes liegt laut den Autoren auf den „harten“ Fähigkeit wie technisches Wissen, wissenschaftliche Managementprinzipien, Nutzung von Werkzeugen und greifbaren Ereignissen, welche für das Managen von Projekten als notwendig erachtet werden. (vgl. Pant & Baroudi, 2006)

Ironischerweise sind die Soft Skills viel schwieriger zu meistern und effektiv zu nutzen, weil sie mit dem EQ (Emotional Quotient) einer Person zusammenhängen. (vgl. Tiago Miranda & Bimal Ghimire, 2007)

Hierbei handelt es sich nämlich um Kompetenzen, welche nicht so leicht in einem Kurs mit nur wenigen Einheiten erlernt werden können. Im Gegensatz zu technischen Fähigkeiten, die eventuell auch im Selbststudium bzw. bei Schulungen am Arbeitsplatz, aufgrund ihres raschen Wandels, erlernbar sind, ist der Erwerb von Soft-Skills ein langfristiger Prozess. (Stevens Matt & Norman Richard, 2016)

Kurse wie *„Kommunikation und Teamarbeit für InformatikerInnen“* an der Universität Wien, welche Theorien zu Soft-Skills vermitteln und diese auch einüben, sind leider noch selten, könnten aber Studierenden helfen, für die Anforderung am Arbeitsmarkt gerüstet zu sein.

Denn auch „weiche“ Qualifikationen können erlernt und trainiert werden. Dies sollte jedoch nicht nur nebenläufig bei Projektarbeiten oder dergleichen geschehen, denn die bewusste Auseinandersetzung mit den dazugehörigen Theorien ist essenziell um diese Fähigkeiten tatsächlich verbessern zu können.

Kommunikation im Bezug zum Lehrplan Informatik

Der § 2 des Österreichischen Schulunterrichtsgesetzes führt unter anderem als Aufgabe der Schule an, dass die Schülerinnen und Schüler *„[...] für das Leben und den künftigen Beruf [...]“* mit dem *„[...] erforderlichen Wissen und Können auszustatten und zum selbstständigen Bildungserwerb zu erziehen“* sind, sowie *„[...] zu selbstständigem Urteil und sozialem Verständnis geführt [...]“* werden. (vgl. „RIS - Schulorganisationsgesetz § 2 - Bundesrecht konsolidiert,“)

Im Lehrplan der AHS-Oberstufe wird an mehreren Stellen Bezug zu den auszubildenden sozialen Fähigkeiten genommen.

Im allgemeinen Teil steht: *„[...] Schülerinnen und Schüler Fähigkeiten erwerben, die später in Ausbildung und Beruf dringend gebraucht werden, etwa für die Bewältigung kommunikativer und kooperativer Aufgaben“*. („RIS - Lehrpläne – allgemeinbildende höhere Schulen - Bundesrecht konsolidiert, Fassung vom 18.03.2018,“). Es gibt einen speziellen Unterpunkt für den *„Bildungsbereich*

Sprache und Kommunikation“, in dem darauf hingewiesen wird, dass die Kommunikationsfähigkeiten der Schülerinnen und Schüler dahingehend auszubilden sind, dass sie in der Lage sind „[...] ihre kognitiven, emotionalen, sozialen und kreativen Kapazitäten zu nutzen und zu erweitern“.

Im Abschnitt *„Bildungsbereich Mensch und Gesellschaft“* wird auf die Vorbereitung zur Teilhabe am öffentlichen Leben und zu sozialen Zusammenhalt und der Notwendigkeit Sachkompetenzen durch Selbst- und Sozialkompetenzen zu erweitern, eingegangen. Ein weiteres Ziel *„des Lernens in der Schule [...]“* ist „[...] die Fähigkeit und Bereitschaft, Verantwortung zu übernehmen, mit anderen zu kooperieren [...] und an der Gestaltung des sozialen Lebens [...] mitzuwirken“.

Im Abschnitt *„Bildungsbereich Kreativität und Gestaltung“* wird auf die Wichtigkeit des Ausdrückens von Gedanken und Gefühlen in verbaler und nonverbaler Form hingewiesen und dass den Schülerinnen und Schülern die Möglichkeit gegeben werden soll, „[...] individuelle Fähigkeiten zu entdecken [...] und sich mit den Ausdrucksformen ihrer Mitmenschen auseinander zu setzen.“

Weiters wird im Abschnitt *„Allgemeine Didaktische Grundsätze - Mehrsprachigkeit“* festgehalten, dass der kompetenzorientierte Unterricht „[...] an den Prinzipien der kommunikativen Didaktik auszurichten“ und die „[...] Förderung und Weiterentwicklung sprachlicher Fähigkeiten [...] von besonderer Bedeutung“ ist. (ebd.)

Zu den Fachkompetenzen im Lehrplan für Informatik der 5. Klasse AHS gehört unter anderem die Fähigkeit Kommunikationstechnologien einzusetzen um kooperative und kommunikative Arbeitsweisen umzusetzen. Es ist in allen Bildungsbereichen die Erweiterung von *Sach-, Selbst- und Sozialkompetenz* in den Mittelpunkt zu stellen. Auch die *„[...] Gestaltung eines angenehmen und erfolgreichen Lernklimas beruht auf Vertrauen, auf der Förderung der individuellen Stärken und des kreativen Potenzials“*. (ebd.)

Der personenzentrierte Ansatz nach Carl Rogers

*„[...] the curious paradox is
that when I accept myself
just as I am, then I change“.*
(Rogers, C.R., 1961/1985, p. 17)

Grundlagen

Carl Ransom Rogers (1902-1987) war ein Psychologe und Psychotherapeut und ist der Begründer des personenzentrierten Ansatzes, welchen er erstmals bei einem Vortrag an der Universität von Minnesota im Jahre 1940 vorstellte. Sein Konzept stellt den Mensch und dessen emotionale Aspekte und nicht das Problem in den Vordergrund. Er sieht den Therapeuten als eine Person die dem Klienten bzw. der Klientin hilft persönlich zu wachsen. (vgl. Groddeck, N., 2006, p. 86)

Laut Rogers beinhaltet eine wirksame Beratung eine

„[...] Beziehung, die dem Klienten ermöglicht zu einem Verständnis seiner selbst in einem Ausmaß zu gelangen, das ihn befähigt, aufgrund dieser neuen Orientierung positive Schritte zu unternehmen“.
(vgl. Carl R. Rogers, 1942/1972, p. 28).

Er ist der Meinung, dass der Mensch fähig ist sich konstruktiv weiterzuentwickeln, wenn er ein passendes zwischenmenschliches Klima vorfindet. (vgl. Motschnig Renate & Nykl Ladislav, 2013, p. 22)

Die Grundlage des personenzentrierten Ansatzes ist die **Aktualisierungstendenz**. Darunter versteht man das Streben nach Selbstentfaltung, Wachstum, Autonomie, zu differenzieren und konstruktive Werthaltungen zu entwickeln, sowie die Gewährwerdung des eigenen Seins und Handelns. (vgl. Keil Wolfgang W. & Stumm Gerhard, 2013, p. 10 ff.) Rogers schreibt:

„[...] der Organismus hat eine Grundtendenz und das Bestreben, den Organismus zu verwirklichen, zu erhalten und zu fördern“. (Rogers, C.R., 1951/1995)

Es handelt sich hierbei nicht um ein psychologisches Konstrukt, sondern zählt zum Fundament des Ansatzes und beschreibt dessen Auffassung von der Natur des Menschen.

Nach Rogers Überzeugung hat das Selbst eine Struktur, die sich ständig ändert. Das Selbst betrachtete er als eine Konfiguration, in der die Veränderung eines kleinen, unbedeutenden Aspekts zu einer totalen Rekonstruktion des ganzen Musters führen kann. (Motschnig, Renate & Nykl Ladislav, 2003)

Die **Selbstaktualisierungstendenz** ist ein Teilaspekt der Aktualisierungstendenz und dient dazu um Selbstkonzept zu erhalten und ebenfalls zu entfalten. Das Selbstkonzept eines Menschen, also, das Bild das jemand von sich selbst hat, entsteht durch Erfahrung und das Wahrnehmen von Beziehungen, sowie deren Bewertung und ist ständig im Wandel. Das Bedürfnis nach Wertschätzung und Anerkennung durch andere hat große Bedeutung.

Sind Erfahrungen und Gefühle des Menschen inkongruent mit seinem Selbstkonzept, das heißt sie stimmen nicht mit diesem überein, kommt es zu Störungen und die Selbstaktualisierungstendenz wird blockiert. Diese **Inkongruenz** erzeugt Angst oder Gefühle der Leere, Schuld, Minderwertigkeit und es kommt zu Selbstzweifel, Abwehrprozessen, Depressionen und dergleichen. (vgl. Keil Wolfgang W. & Stumm Gerhard, 2013, p. 10 ff.)

Als eine, über die Psychotherapie hinausgehende Theorie etablierte sich der personenzentrierte Ansatz vermutlich durch die eigene personenzentrierte Kommunikationstheorie. (Rogers, C.R., 1959)

Basisvariablen für personenzentrierte Kommunikation

Die wesentlichen therapeutischen Grundeinstellungen für eine gute zwischenmenschliche Beziehung sind Kongruenz, Empathie und Akzeptanz. Diese Grundhaltungen können laut Rogers eine Atmosphäre schaffen in welcher das andere Individuum

- Aspekte von sich selbst erfahren und verstehen wird, die er zuvor unterdrückt hat
- sich besser integriert fühlen und besser in der Lage sein wird, effektiv zu funktionieren
- jener Person, die er sein möchte ähnlicher werden wird
- selbstbestimmter, selbstbewusster und einzigartiger sein wird
- von anderen mehr Akzeptanz und Verständnis bekommen wird
- besser in der Lage sein wird Probleme des Lebens adäquater und besser zu bewältigen (vgl. Rogers, C.R., 1961/1985, p. 37 ff.)

Kongruenz (Echtheit):

*„Wir sind dann echt oder kongruent,
wenn unser inneres Erleben
mit der Wahrnehmung dieses Erlebens
und der Kommunikation übereinstimmen.“
(Motschnig Renate & Nykl Ladislav, 2013, p. 100)*

Die Therapeutin bzw. der Therapeut ist sich dessen, was er erlebt bewusst und kann es in der Beziehung mit der Klientin bzw. dem Klienten leben und gegebenenfalls kommunizieren. (vgl. Rogers, C.R., 1980, p. 115 ff.)

Der Eindruck der dem Gegenüber vermittelt wird muss sich mit der verbalen Ausdrucksweise decken. Es ist wichtig, dass Erfahrung und Bewusstsein zusammenpassen und der Ausdruck ehrlich, offen und transparent ist. (vgl. Motschnig Renate & Nykl Ladislav, 2013, p. 37 ff.).

Dieser Zustand gliedert sich in zwei Aspekte. Einerseits ist es die Übereinstimmung zwischen dem inneren Erleben und dem bewussten Zugang dazu, andererseits die getreue Kommunikation darüber. Rogers ist der Meinung, dass vollkommene Kongruenz in allen Situationen kaum erreichbar ist und es sich um einen immerwährenden Prozess handelt. (ebd., p. 98 ff.).

Akzeptanz (positive Beachtung)

*„Je mehr ich den Einzelnen zu akzeptieren vermag,
[...] desto leichter kann ich
eine für ihn nützliche Beziehung schaffen.“
(Rogers, C.R., 1973/2009, p. 47)*

Eine bedingungslose Wertschätzung der anderen Person, die dieser ermöglicht sich nicht zu verstellen, nichts vorgeben oder verbergen zu müssen. Es besteht dabei jedoch nicht die Notwendigkeit, dass man deren Meinung teilt oder ihrem Verhalten zustimmt. Allerdings bedeutet es ihrer Richtung zu vertrauen und ihr eigenen Freiraum und Entwicklungsspielraum zugestehen, ohne die eigenen Reaktionen zu unterdrücken. Das bedeutet, dass Akzeptanz nur in Kombination mit Kongruenz und empathischem Verstehen bestehen kann. (vgl. Motschnig Renate & Nykl Ladislav, 2013, p. 107 ff.)

Sie kann vermittelt werden, indem eine Person in Aktivitäten miteinbezogen, ihr Raum gegeben und sie eingeladen wird, sowie ihr Vertrauen und aufrichtige Gefühle vermittelt werden. (vgl. Motschnig Renate & Nykl Ladislav, 2013, p. 111)

Empathie (Einführendes Verstehen)

*„Die Erfahrung, verstanden zu werden,
ist an sich schon entwicklungsfördernd!“*

(Pörtner, M., 1996, p. 29)

Empathie bedeutet echtes Verständnis und Einfühlungsvermögen. Rogers hält dies nicht für einen Zustand, sondern eher für einen Prozess der:

„[...] bedeutet, die private Wahrnehmungswelt des anderen zu betreten und darin ganz und gar heimisch zu werden, [...] bedeutet zeitweilig das Leben dieser Person zu leben, sich vorsichtig darin zu bewegen, ohne vorschnell Urteile zu fällen“. (Rogers, C.R., 1980, p. 79)

Es bedeutet, der anderen Person zuzuhören und zu versuchen sie zu verstehen und ihre Gefühlswelt wahrzunehmen. Es ist auch die Fähigkeit durch Einfühlsamkeit eine Situation besser wahrzunehmen, mehr zu erfahren, wodurch es möglich wird besser zu verstehen, aber auch mehr über sich selbst zu erfahren. Es ist unbedingt erforderlich das entgegengebrachte Vertrauen zu hüten. (vgl. Motschnig Renate & Nykl Ladislav, 2013, p. 113 ff.)

Um Empathie ausstrahlen zu können ist aktives Zuhören essenziell. (siehe Aktives Zuhören)

Ein Zusammenspiel dieser Basisvariablen ist für eine personenzentrierte Kommunikation unverzichtbar. Dies verdeutlicht Rogers folgendermaßen:

„Damit Therapie in Gang kommt, ist die Ganzheit (Kongruenz) des Beraters in der Beziehung von primärer Bedeutung, jedoch sollte ein Teil seiner Kongruenz in der Erfahrung der bedingungslosen positiven Beachtung und des empathischen Verstehens bestehen“. (Rogers C.R. 1959/1991, p.42) zitiert in (vgl. Motschnig Renate & Nykl Ladislav, 2013, p. 121)

Aktives Zuhören

*„Rückblickend wird mir klar, daß [sic] mein Interesse am Gespräch
[...] zum Teil wohl aus meiner früheren Einsamkeit herrührte.
Hier gab es einen [...] Weg, Menschen wirklich nahe zu kommen
und vielleicht ein wenig den Hunger nach Kommunikation zu stillen [...]“
(Rogers, C.R. & Rosenberg Rachel L., 1980, p. 189)*

Aktives Zuhören wurde von Rogers und Farson entwickelt und ist eine Art des Zuhörens und des auf eine andere Person Reagierens, die dazu beiträgt das gegenseitige Verständnis zu verbessern unter Verwendung verbaler und nonverbaler Kommunikationsfähigkeiten. (vgl. Rogers, C.R., 1980)

Beim Aktiven Zuhören handelt es sich nicht um eine Kommunikationstechnik, sondern um eine grundlegende Einstellung der Zuhörerin oder des Zuhörers, welche es ermöglicht aktives und echtes Interesse der sich mitteilenden Person zu erleben und auch zu vermitteln wie auch deren Blickwinkel zu erkunden und zu berücksichtigen.

Das Aktive Zuhören bewirkt eine Veränderung bei der Person der zugehört wird. Diese reift emotional, ist weniger defensiv und autoritativ, wird demokratischer und ist selbst eher bereit besser zuzuhören. Sie neigt darüber hinaus dazu, sich selbst besser zuzuhören und sich ihrer Gefühle und Gedanken besser bewusst zu werden. Aber auch bei der zuhörenden Person entsteht ein Wachstumserlebnis, da sie nicht nur mehr Informationen erhält, sondern auch ihre Einstellung konstruktiv verändert wird. (vgl. Motschnig Renate & Nykl Ladislav, 2013, p. 59 ff.)

Wichtig beim aktiven Zuhören ist zu erfassen was die andere Person aus ihrer Sicht kommuniziert und darüber hinaus zu vermitteln die Dinge aus deren Sicht zu sehen. Es erfordert totale Aufmerksamkeit und Reaktion auf die zum Ausdruck kommenden Gefühle. Um einen Menschen davon zu überzeugen, dass er hörens-wert ist und respektiert wird, bedarf es eines entsprechenden Verhaltens. Worte alleine können dieses Gefühl kaum vermitteln. (vgl. Rogers, C.R. & Farson, R.E., 1957)

Da sich die Person welcher zugehört wird angenommen und verstanden fühlen soll, sind einige Faktoren und Techniken welche aktives Zuhören begünstigen zu beachten.

Es ist wesentlich, dass die Person erfährt, dass ihr die volle Aufmerksamkeit zuteil wird. Techniken welche dazu beitragen sind das **Paraphrasieren, Verbalisieren und nonverbale sowie verbale Aufmerksamkeitssignale**.

Um **Aufmerksamkeit** zu signalisieren, eignen sich folgende nonverbale und verbale Aktionen: Blickkontakt, zustimmendes Kopfnicken, konzentrierter Gesichtsausdruck, dem Sprecher zugewandte, offene Körperhaltung und verbale Reaktionen wie zum Beispiel ein „hm“, „aha“, „interessant“ oder dergleichen.

Das **Paraphrasieren** dient dazu, über das akustische Verstehen hinaus, die Bedeutung mit dem sinnlich Wahrgenommenen zu verknüpfen. Dies ist dadurch, dass jede Nachricht vier Ebenen aufweist nicht automatisch gegeben. Friedemann Schulz von Thun betrachtet den Vorgang zwischenmenschlicher Kommunikation gleichsam von vier Seiten: *„dem Sachaspekt, dem Beziehungsaspekt, dem Selbstoffenbarungsaspekt und dem Appellaspekt.“* Dies hat er mit seinem *„Kommunikationsviereck“* verdeutlicht. Daraus ergibt sich, dass ein und dieselbe Nachricht viele Botschaften enthält und es dadurch dem Empfänger der Nachricht erschwert die tatsächliche Bedeutung auch wirklich zu verstehen. (vgl. Schulz von Thun, F., 1981, p. 14 ff.)

Beim Paraphrasieren wird das Gesagte mit eigenen Worten wiederholt und mittels Nachfragen ergänzt, um das richtige Verständnis sicherzustellen. (vgl. Bay, R.H., 1998, p.4 ff.) Dadurch wird einerseits Interesse gezeigt und andererseits können Missverständnisse vermieden werden.

Beim **Verbalisieren** werden hingegen die offenen oder versteckten Gefühle des Gesprächspartners angesprochen und somit an den drei emotionalen Aspekten des *„Kommunikationsvierecks“*, also dem *„Beziehungsaspekt“*, dem *„Selbstoffenbarungsaspekt“* und dem *„Appellaspekt“* angeknüpft.

Formen des Verbalisierens sind weiterführende Fragen nach Gefühlen, klärende Fragen bezüglich scheinbaren Nebensächlichkeiten und Statements, welche die Gefühle des Gegenübers benennen. (vgl. Bay, R.H., 1998, p. 64 ff.)

Aktives Zuhören erfordert allerdings Mut, da durch wirklich gute Kommunikation möglicherweise eigene Einstellungen adaptiert werden müssen und man Gefahr läuft sich selbst zu verändern. (vgl. Rogers 1973/2009; p. 325)

Nach Rogers und Farson gibt es einiges, was den Aufbau eines förderlichen Gesprächsklimas beeinträchtigen kann. Hierzu gehört das Beurteilen. Egal ob dieses Urteil positiv oder kritisch ist, denn dadurch wird das Bilden einer eigenen Ansicht gehemmt und eine Weiterentwicklung verhindert. Ratschläge, Ermunterungen und Informationsangebote sollten ebenfalls vermieden werden, da mit diesen versucht wird eine Person zu beeinflussen.

Weiters führen festgefahrene Denkmuster, fehlendes Vertrauen, das Bedürfnis sich selbst mitteilen, erklären oder rechtfertigen zu wollen zu Problemen in der Kommunikation.

Es ist jedoch keineswegs notwendig allem zuzustimmen, sondern lediglich die wechselseitige Akzeptanz und das Verstehen zu fördern. (vgl. Motschnig Renate & Nykl Ladislav, 2013, p. 108 ff.)

Eine Beobachtung, welche auf der Tatsache beruht, dass das Selbst eine Struktur hat, welche sich laufend verändert, ist, dass Reorganisationen des Selbst zu jedem Zeitpunkt während oder nach einem personenzentriertem Gespräch stattfinden können. (Motschnig, Renate & Nykl Ladislav, 2003)

Aufgrund der Wichtigkeit des Aktiven Zuhörens für eine personenzentrierte Kommunikation und signifikantes Lernen ist im Unterrichtskonzept dieser Arbeit vorgesehen, dass dieses Thema mit den Schülerinnen und Schülern explizit bearbeitet wird. (siehe Anhang)

Das Humanistische Menschenbild

„Wenn ich Menschen nicht dazwischenfahre, passen sie auf sich selbst auf.“

„Wenn ich Menschen nicht befehle, verhalten sie sich von selbst richtig.“

„Wenn ich Menschen nicht , werden sie von selbst besser.“

„Wenn ich mich Menschen nicht aufdränge, werden sie sie selbst“

(Rogers, C.R. & Rosenberg Rachel L., 1980, p. 196)

Die humanistische Psychologie wurde unter anderem von Abraham Maslow, Carl Rogers, Charlotte Bühler, Rollo May, Gordon Allport, Erich Fromm, Ruth Cohn sowie Fritz und Lore Perls ab Mitte des 20. Jahrhunderts entwickelt und wird neben der Psychoanalyse und dem Behaviorismus oft als die *„Dritte Kraft in der Psychologie“* bezeichnet. (vgl. Hinte; W., 1980, p.48)

Einige **Grundannahmen** des Humanismus lauten:

- Der Mensch ist mehr als die Summe seiner Teile – er kann nicht auf Komponenten reduziert werden. Obwohl es wichtig ist Kenntnisse über die Teilfunktionen einer Person zu haben, steht deren Einzigartigkeit und Mensch-Sein im Vordergrund.
- Der Mensch lebt in zwischenmenschlichen Zusammenhängen. Die o.a. Einzigartigkeit ist an Beziehungen zu anderen Menschen gebunden.
- Der Mensch ist sich seiner bewusst. Ein Wesensmerkmal des Menschen und die Grundlage seines Verstehens ist sein Bewusstsein – unabhängig davon wie viel davon ihm zugänglich ist.
- Der Mensch ist fähig zu entscheiden und sich aus eigenem Antrieb zu verändern. Er kann aktive Entscheidungen treffen, da er sich seiner bewusst ist.
- Der Mensch ist zielgerichtet und strebt nach Bedeutung.

Die *„Humanistische Psychologie stellt den Menschen in den Mittelpunkt“* und fordert, dass auch der forschende Mensch selbst immer Teil der Forschung sein muss. Nicht die Methode steht im Vordergrund, sondern die Bedeutungszusammenhänge des menschlichen Seins. Die Methoden sollen der menschlichen Erfahrung untergeordnet werden. Auch akzeptiert sie die Relativität des Wissens und postuliert ein Universum unendlicher Möglichkeiten. Sie vertraut weitestgehend der phänomenologischen Orientierung und versucht andere Orientierungen zu ergänzen, ohne deren Leistungen zu schmälern. (vgl. Bugental, James F. T, 1964, p. 23 ff.)

Rogers vertritt laut Motschnig und Nykl die Ansicht ,

„[...] dass der menschliche Organismus vertrauenswürdig ist [...] und eine reiche Anlage vorhanden ist, die er dazu freisetzen kann sich selbst zu verstehen und konstruktiv zu verändern.“ (Aktualisierungstendenz) (Motschnig Renate & Nykl Ladislav)

Humanistische Pädagogik - Der personenzentrierte Unterricht

*„[...] den Lernenden das Lernen zu erleichtern
und echte zwischenmenschliche Beziehungen aufzubauen [...]“
(Motschnig Renate, 2010)*

Carl R. Rogers entdeckte Bedingungen gelten nicht nur in der Psychotherapie und im Gespräch zwischen zwei Menschen. Sie sind vor allem auch in Unterrichtssituationen relevant. (vgl. Motschnig Renate & Nykl Ladislav).

Er verfasste zahlreiche Artikel zu einer personenzentrierten Sichtweise des Unterrichts und veröffentlichte 1969 eine Sammlung dieser Werke, welche er mit Erfahrungen und Erkenntnissen ergänzte, unter dem Titel *„Freedom to Learn“* (vgl. Barrett-Lennard, Godfrey T, 1998, p. 185)

Rogers Hypothesen in Bezug auf Lernsituationen:

- *„Wir können einer anderen Person nicht direkt etwas lehren; wir können nur ihr Lernen fördern.“*
- *„Die Struktur und Organisation des Selbst lockert sich, nur wenn jegliche Bedrohung fehlt, ansonsten wird sie starr.“*
- *„Die Bildungssituation, die das signifikante Lernen am effektivsten fördert, ist eine, in der
(1) die Gefahr für das Selbst des Lernenden auf ein Minimum reduziert wird und
(2) differenzierte Wahrnehmung des Erfahrungsfeldes erleichtert wird.“
(Rogers 1951/1995: 389-391 zitiert in: Barrett-Lennard, Godfrey T, 1998, p. 184)*

Rogers beschreibt wachstumsfördernde Grundhaltungen für den Unterricht wie folgt:

- Der Lehrende hat ausreichend Selbstbewusstsein, um darauf zu vertrauen, dass die anderen selbst denken und lernen können.
- Die Verantwortung für den Lernprozess teilen sich Lehrende, Lernenden und deren Umfeld.
- Der Lehrende schafft Lernmöglichkeiten aufgrund seiner Erfahrung und seiner Person und ermutigt die Lernenden, Lernmöglichkeiten selbst beizusteuern.
- Der Lernende entwickelt alleine oder gemeinsam mit anderen sein eigenes Lernprogramm.
- Es wird mithilfe von Authentizität, verstehenden Zuhörens und des Füreinander-Besorgtseins ein förderliches Lernklima geschaffen.
- In erster Linie wird der andauernde Lernprozess gefördert.
- Der Lernende erkennt und akzeptiert, dass es Selbstdisziplin erfordert, um sein Ziel zu erreichen.

- Der Lernende beurteilt den Umfang und die Bedeutung seines Lernens selbst. (vgl. Rogers, C.R. & Rosenberg Rachel L., 1980, p. 133 ff.)

Ergänzend dazu beschreiben Renate Motschnig-Prink und Michael Derntl:

- Der Lehrende übernimmt die Funktion eines Moderators, der die Bedingungen des personenzentrierten Ansatzes erfüllt.
- Der Lernende beteiligt sich an allen Aspekten des Lernens und der Entscheidungsfindung
- Lernende können Themen, Aspekte und Prozesse im Kontext vorschlagen oder auswählen
- Konventionelle Elemente haben ihren Platz, wenn sich die Gruppe dafür entscheidet.
- Das Lernen hat mehrere Dimensionen. Es ist sowohl individuell als auch kooperativ, sowohl Inhalts- als auch Prozessorientiert, sowohl kognitiv als auch persönlich, sozial, intuitiv oder emotional.
- Jeder wird gehört, alle Ressourcen und Einschränkungen werden berücksichtigt. (vgl. Motschnig-Pitrik & Derntl, Michael, 2002)

Rogers ist der Meinung:

„[...] dass in einem solchen wachstumsfördernden Klima das Lernen tiefer wird, schneller voran geht und größeren Einfluss auf das Verhalten und Leben des Lernenden hat als der herkömmliche Unterricht.“
(Rogers, C.R. & Rosenberg Rachel L., 1980, p. 133 ff.)

Die drei Basisvariablen aus Rogers personenzentriertem Ansatz in der Psychotherapie, gelten also auch für die Begegnung zwischen Lernenden und Lehrenden.

Anne-Marie und Reinhard Tausch nennen folgende notwendigen und hinreichenden Bedingungen:

Achtung - Wärme und Rücksichtnahme von Person zu Person:

- „den anderen wertschätzen, an ihm teilnehmen“
- „ihm Geltung schenken, ihn anerkennen, ihn willkommen heißen, ihm zugeneigt sein“
- „mit ihm freundlich, herzlich umgehen, mit ihm nachsichtig sein“
- „ihn rücksichtsvoll, zärtlich, liebevoll behandeln“
- „ihn ermutigen, ihn wohlwollend behandeln“
- „ihm vertrauen“
- „zu ihm halten, ihm beistehen, ihn beschützen, ihn umsorgen, ihm helfen, ihn trösten“
- „sich ihm gegenüber öffnen, ihm nahe sein“ (Tausch, Reinhard & Tausch, Anne-Marie, 1991, p. 120)

Vollständiges – einführendes Verstehen versucht die andere Person so zu erfassen, wie sich diese selbst wahrnimmt

- „eine Person erfasst vollständig die vom anderen geäußerten gefühlsmäßigen Erlebnisinhalte [...]“
- „sie wird gewahr, was die Äußerungen oder das Verhalten für das Selbst des anderen bedeuten“
- „sie versteht den anderen so, wie dieser sich im Augenblick selbst sieht“
- „sie teilt dem anderen das mit, was sie von seiner inneren Welt verstanden hat“
- „sie hilft dem anderen, die von ihm gefühlten Bedeutungen dessen zu sehen, was er geäußert hat“
- „sie ist dem anderen in dem nahe, was dieser fühlt, denkt und sagt“
- „sie zeigt in ihren Äußerungen und Verhalten das Ausmaß an , inwieweit sie die Welt des anderen mit seinen Augen sieht“
- „sie drückt die vom andern gefühlten Inhalte und Bedeutungen in tiefgreifender Weise aus als dieser es selbst konnte“
- „ihre Handlungen und Maßnahmen sind dem persönlichen Erleben des anderen angemessen“ (ebd., p. 181)

Echtheit - Aufrichtigkeit

- „eine Person das sagt, was sie denkt und fühlt“
- „sich so gibt, wie sie wirklich ist“
- „sich ungekünstelt verhält, natürlich, spielt keine Rolle“
- „sie ist ohne professionelles, routinemäßiges Gehabe“
- „sie ist sie selbst, sie lebt ohne Fassade und Panzer“
- „sich verhält sich in individueller, origineller, vielfältiger Weise“
- „sie ist vertraut mit dem, was in ihr vorgeht“
- „sie ist aufrichtig und heuchelt nicht“
- „sie ist ehrlich sich selbst gegenüber, macht sich nichts vor, ist bereit, das zu sein, was sie ist“
- „sie offenbart sich anderen und gibt sich mit ihrem Ich zu erkennen, sie verleugnet sich nicht “
- „sie ist durchsichtig“
- „sie drückt tiefe gefühlsmäßige Erlebnisse aus“ (ebd., p. 215)

Diese drei Grundhaltungen sind für pädagogische Prozesse entscheidend. Sie tragen dazu bei, dass sich Schülerinnen und Schüler verstanden fühlen und eine Offenheit zum Weiterdenken entwickeln können die es ihnen ermöglicht vorhandene Muster zu hinterfragen und gegebenenfalls zu durchbrechen.

In einem solchen Lernklima, das eine bewusste Wahrnehmung des Selbst, anderer und der Umgebung ermöglicht, bietet sich die Gelegenheit aktualisiert zu werden, anstatt potenziell zu bleiben. Diese zeigt sich in der Fähigkeit die Aspekte des Lebens und seiner selbst zu verstehen, auch solche die bis dahin verborgen waren. (vgl. Rogers, C.R., 1973/2009, p. 49)

Konsequente Forschung (Aspy, 1972; Cornelius-White, 2003; Rogers, 1961; Tausch & Tausch, 1998) im Bereich des schülerzentrierten Ansatzes bewiesen, dass Lernende bessere Ergebnisse und persönliches Wachstum in Form von höherem Selbstvertrauen, Kreativität, Offenheit für Erfahrung, Selbstachtung und Respekt gegenüber anderen und ihrer Umwelt usw. erzielen, wenn sie in einer Atmosphäre oder einem Klima, in welcher der Lehrende die oben angeführten drei zentralen Einstellungsbedingungen hat und wenn sie diese zumindest bis zu einem gewissen Grad wahrnehmen. (vgl. Motschnig-Pitrik, R. & Malich, K., 2004)

Lernender zentrierter Unterricht gewinnt immer mehr an Bedeutung, da sich das Bildungsziel immer mehr hin zum Aufbau von Kompetenzen, zur Übernahme von Verantwortung, zu effektiver Arbeit in Teams und zur Förderung von Kreativität verlagert. (Motschnig, Renate, Sedlmair, Michael, Schroder, Svenja, & Möller, Torsten, 2016)

Er besitzt ein enormes Potential:

„[...] da er auf die individuelle Person selbst abgestimmt ist: auf deren geerbte Eigenschaften, das Erlernte, ihre Gefühle und Werte und -- allem voran -- ihre eigene Erfahrung. Darüberhinaus hilft der Ansatz nicht nur, Probleme zu lösen, sondern ist insbesondere auch geeignet, Persönlichkeitswachstum in Richtung höherer Kreativität, größerer Offenheit und Ausdruckskraft, größerer innerer Ausgeglichenheit, höherer Selbstbestimmung und Selbstzufriedenheit, etc. zu fördern“.

(Rogers 1995 zitiert in: Motschnig Renate & Nykl Ladislav).

Es sind genau solche Eigenschaften welche in der Berufswelt aber im speziellen in der Informatik von großer Bedeutung sind. (Motschnig Renate & Nykl Ladislav)

Signifikantes Lernen

*„Es ist ein Lernen, das etwas ändert
- im Verhalten des einzelnen,
in den von ihm zukünftig einzuschlagenden Handlungsweisen,
in seinen Einstellungen und in seiner Persönlichkeit“.
„Es ist ein durchdringendes Lernen
[...] etwas, das jeden Teil seiner Existenz betrifft und durchdringt“.
(Rogers, 1973/2009, S. 273 f.).“*

Signifikantes Lernen ist bedeutungsvolles Lernen. Es erfasst die ganze Person, sowohl die rationalen als auch seine emotionalen Anteile und geht über reine Fakten- und Wissensansammlung weit hinaus. Dabei wird der Mensch mit seinem Wissen, seinen Fähigkeiten und seinen Emotionen in das Zentrum der Überlegungen gerückt. (Motschnig-Pitrik, R., 2005)

Das Lernen wird als bedeutungsvoll und zweckmäßig erlebt und ist ein direkter Ausdruck der Aktualisierungstendenz aus Rogers personenzentrierter Theorie. *„In conceptual terms, the process is a direct expression of the individual's actualizing tendency“* (Barrett-Lennard, Godfrey T, 1998, p. 187)

Studien haben ergeben, dass in personenzentrierter Therapie signifikantes Lernen stattfindet das tatsächlich etwas ändert:

- *„Der Mensch kommt dazu, sich anders zu sehen“*
- *„Er akzeptiert in stärkerem Maße sich und seine Gefühle“*
- *„Er wird selbstbewusster und selbstbestimmter“*
- *„Er wird in seinen Wahrnehmungen flexibler [...]“*
- *„Er setzt sich realistischere Ziele“*
- *„Sein Verhalten ist reifer“*
- *„Er ändert unangepasste Verhaltensweisen [...]“*
- *„Er wird gegenüber anderen akzeptierungsbereiter.“*
- *„Er wird offener für die Bedeutung dessen, was sich außerhalb wie innerhalb seiner selbst abspielt.“*
- *„Er verändert sich auf konstruktive Art und Weise in seinen grundlegenden Persönlichkeitseigenschaften.“* (Rogers, C.R., 1973/2009, p. 274)

Rogers bezeichnet dieses Lernen als signifikant, da es Veränderung bedeutet - im Verhalten, in Zusammenhang mit zukünftigen Handlungsweisen und Einstellungen, in der Persönlichkeit. (vgl. ebd.)

Das neue Wissen wird also nicht nur gespeichert. Es bedeutet eine Bewusstseinsveränderung und die beeinflusst nachfolgende Wahrnehmungen sowie das Verhalten des Lernenden. Die Übereinstimmung von Zweckmäßigkeit und Wirkung des Lernens ist für den Lernenden befriedigend und steigert sein Selbstbewusstsein. Auch werden beim Lernenden die Neugierde, das Verlangen zu wissen und zu entdecken und neue Fähigkeiten geweckt. Die Entwicklung des Lernenden und somit auch seine Originalität wird so gefördert, bringt das kreative Potenzial des Einzelnen zur Geltung und erhöht seine Fähigkeit und Bereitschaft Probleme zu lösen. (vgl. Barrett-Lennard, Godfrey T, 1998, p. 187 ff.)

Diese Form des Lernens ist nachhaltiger, als das Ansammeln von intellektuellem Wissen, welches oft nicht als zusammenhängend wahrgenommen wird, da neben der kognitiven Ebene auch Gefühle, Emotionen, Intuition und soziale Kompetenzen mit einbezogen sind. (Motschnig-Pitrik, R. & Mallich, Katherina, 2002)

Auf einen Zusammenhang zwischen Kognition und Emotion wird auch in den Neurowissenschaften hingewiesen:

„Associative learning has linked emotions with thoughts in a rich two-way network. Certain thoughts evoke certain emotions and vice versa. Cognitive and emotional levels of processing are continuously linked in this manner.“ (Damasio, A., 2000, p. 71) zitiert in: (Motschnig-Pitrik, Renate, 2006, p. 86)

Lernen ist am nachhaltigsten, wenn es alle drei Ebenen der Entwicklung also der Intellekt, die sozialen Fähigkeiten und Einstellungen angesprochen werden. (Motschnig, Renate et al., 2016)

Der Informatikunterricht eignet sich ganz besonders für personenzentrierten Unterricht da dieser:

- *„eigeninitiativ, anders ausgedrückt, selbstgesteuert ist“,*
- *„auf die Lösung realer, authentischer Probleme abzielt“,*
- *„mit interpersonalen Beziehungen, die das Klima prägen, eng verknüpft ist und folglich nur in Beziehungen mit anderen erfolgen kann“,*
- *„durch ein reichhaltiges, breites Angebot an Informationen und Möglichkeiten unterstützt wird“,*
- *„zwar einer Art des Seins entspringt, sich jedoch in zielgerichtetem Verhalten, also auch Handeln, äußert.“ (Motschnig-Pitrik, R., 2004)*

Die Funktion der Lehrenden

*„[...] das Ziel des Lehrenden ist, fortlaufend behilflich zu sein,
die gegensätzlichen und vage formulierten individuellen
Absichten ans Licht zu bringen [...] sich an der Förderung dieser Absichten beteiligen,
indem er auf Hilfsmittel hinweist,
die den Lernenden unter Umständen nicht bekannt sind,
und er wird die Rolle, die ihm von der Gruppe
zur Förderung ihrer Absichten zugewiesen wird,
entweder eindeutig akzeptieren
oder eindeutig ablehnen.“ (Rogers, Carl R., 1983/2005, p. 347)*

Ein entscheidendes Merkmal für eine gute Beziehung zwischen Lehrenden und Lernenden ist der Grad der Kongruenz des Lehrenden in Bezug auf die Lernenden. Das hängt davon ab wie offen sich die Lehrperson in ihrer ganzen Persönlichkeit präsentiert.

Weiters spielt der Grad, indem der Lehrende Gefühle und Einstellungen, aber auch das Thema und den Problemlösungsprozess, aus Sicht der Lernende wahrnimmt, eine entscheidende Rolle, denn was er fühlt beeinflusst seine Ausdrucksweise.

Ein weiterer Aspekt ist das Ausmaß in dem die Lehrperson tatsächlich positive Gefühle für den Lernenden als Person erlebt und diese auch zeigt und inwiefern sie diesen als eigenständiges Individuum auf seinem ganz eigenen Entwicklungsweg schätzt. Es geht hierbei um Respekt, Interesse und Fürsorge.

Es spielt auch eine große Rolle inwiefern der Lehrende in der Lage ist die Lernenden zu respektieren, ohne diese zu beurteilen. Einen weiteren Faktor stellt die Verfügbarkeit der Lehrende dar, seine Bereitschaft informatives und anderes Wissen, inklusive dessen Bedeutung für ihn selbst, zu äußern, falls dieses erwünscht ist. (vgl. Barrett-Lennard, Godfrey T, 1998, p. 188 ff.)

Zusätzlich zu den Einstellungen der Lehrenden werden die Inhalte so angesprochen, dass er mit den Einstellungen übereinstimmt und die Lernziele unterstützt. Dabei ist folgendes zu beachten:

- Der aktuelle Inhalt muss gut zu den Interessen und Zielen der Lernenden passen, um bedeutend für diese zu sein.
- Alle benötigten Ressourcen müssen so bereitgestellt werden, dass die Lernenden diese jederzeit leicht finden können.
- Der Einsatz von Lernverträgen oder Selbstevaluierungsverfahren um Bedrohungen zu minimieren und Aktivitäten sowie signifikantes Lernen zu fördern ist empfehlenswert.

- Wenn möglich sollen verschiedene interaktive Szenarien zum Einsatz kommen wie etwa Team- oder Gruppenarbeiten aber auch Diskussionsveranstaltungen der ganzen Gruppe. (vgl. Barrett-Lennard, Godfrey T, 1998, p. 183 ff.)

Carl Rogers Erwartungen an einen Lehrenden findet man in seinem Werk „Freedom to Learn“ :

- Der Lehrende schafft ein positives Lernklima.
- Er hilft, Lernziele aufzuzeigen und zu klären.
- Er vertraut, dass der Lernende aktiv lernen will.
- Er stellt umfangreiches Material zur Verfügung.
- Er sieht sich selbst als Medium zum Lernen.
- Er bietet intellektuellen und emotionalen Inhalten den Raum der von der Gruppe oder vom Einzelnen erwartet wird .
- Er wird in einer Atmosphäre der gegenseitigen Anerkennung zum Mitlernenden bzw. zu einem Teil der Gruppe.
- Er bemüht sich seine eigenen Grenzen zu erkennen und akzeptiert diese.
- Er teilt seine Gefühle und Gedanken der Gruppe mit ohne dabei zu urteilen oder etwas zu fordern.
- Er ist aufmerksam bezüglich starker Gefühle der Lernenden bzw. der Gruppe um auf diese eingehen zu können. (vgl. Rogers; Carl Ransom, 1969, p. 163 ff.) (vgl. Rogers, C.R., 1988, p. 176 ff.)

Schülerorientiertes Lernen beginnt, wie Rogers betonte rund um die Ziele der Lernenden. Da es Schülerinnen und Schülern oft schwerfällt eigene Ziele zu formulieren, vor allem denjenigen welche nicht daran gewöhnt sind, ist es eine der Hauptaufgaben des Lehrenden laufend beim Finden individueller Ziele zu helfen und diese zu einem Gemeinschaftszweck zu verbinden.

Der Lehrende stellt eine Quelle der Information über verfügbare Ressourcen, welche für die Erreichung der gemeinsamen sowie der individuellen Ziele relevant sind, dar. So kann ein intrinsisch motivierter, aktiver Suchprozess entstehen, der zu einem sinnvollen, integrativen Lernen führt. (vgl. Barrett-Lennard, Godfrey T, 1998, p. 184)

Auch Reinhard und Anne-Marie Tausch erwähnen die materiellen und menschlichen Hilfsquellen um selbstständiges Arbeiten zu erleichtern. *„Weitgehend selbstständiges, selbstverantwortliches Lernen der Schüler wird wesentlich erleichtert, ja zum Teil erst ermöglicht durch konkretes Anbieten von anregenden Lernmaterialien und Lernhilfen“* (Tausch, Reinhard & Tausch, Anne-Marie, 1991; p. 288 ff.). Dabei ist darauf zu achten, dass die Ressourcen altersgerecht sind und die Lernenden weder über- noch unterfordern. (vgl. Tausch, Reinhard & Tausch, Anne-Marie, 1991, p. 289)

Zum Thema Ressourcen schreibt Rogers, dass, falls eine Lehrperson seine Ansichten vertrete, so würde er sich folgendermaßen zur Verfügung stellen:

- Sie würde ihre Kenntnisse und Spezialerfahrungen auf dem speziellen Inhalt bekannt geben, jedoch nur so, dass die Lernenden dieses Wissen benutzen können aber nicht das Gefühl haben dies auch zu müssen.
- Sie würde einen Vortrag ihres Wissens anbieten. Die Lernenden müssten jedoch selbst entscheiden, ob sie davon Gebrauch machen wollen.
- Sie würde, weitere Ressourcen finden und zur Verfügung stellen, falls die Gruppe dies zur Förderung des Lernprozesses wünscht. (vgl. Rogers, C.R., 2009, p. 282)

Personenzentrierter Unterricht hängt nicht nur von den Plänen des Lehrenden ab. Er birgt das Unerwartete in sich, die Chance aus Situationen im „Hier und Jetzt“ zu lernen. Dazu ist eine große Flexibilität, sowohl von den Lernenden als auch von den Lehrenden erforderlich. Der persönliche Stil, die Erfahrung und die sozialen Fähigkeiten der Lehrperson spielen eine große Rolle. (vgl. Motschnig-Pitrik & Derntl, Michael, 2002)

Um personenzentriert unterrichten zu können muss man dies über einen längeren Zeitraum lernen, Unterstützung von anderen annehmen und mit Stress umgehen können, denn kaum jemand kann diese Haltung ständig einnehmen. Auch stellt der personenzentrierte Unterricht keine Garantie für eine angemessene Entwicklung junger Menschen dar, es ist lediglich eine sehr günstige Bedingung. (vgl. Tausch, Reinhard & Tausch, Anne-Marie, 1991, p. 99)

Keineswegs hat der personenzentrierte Unterricht etwas mit dem „Laissez-Faire“ Stil zu tun. Die Haltung der Lehrperson ist nicht als Schwäche zu interpretieren, denn Grenzüberschreitungen sollen nicht ignoriert, sondern mit im Vorfeld vereinbarten Konsequenzen belegt werden. Hierbei wird jedoch nicht bewertet, beschuldigt oder beschimpft vielmehr wird aufgezeigt, dass das Überschreiten von Grenzen Konsequenzen hat. (vgl. Tausch, Reinhard & Tausch, Anne-Marie, 1991, p. 9 ff.)

Die Funktion der Lernenden

*„Der einzige Mensch, den man gebildet nennen kann,
ist jener, der gelernt hat, wie man lernt;
der gelernt hat, wie man sich anpasst und ändert;
der erkannt hat, dass kein Wissen sicher ist,
dass nur der Prozess der Suche nach Wissen eine Basis für Sicherheit bietet“.*
(Rogers; Carl Ransom, 1969, p. 105 ff.)
zitiert in (Motschnig Renate & Nykl Ladislav, 2013, p. 176)

Bestimmt der Lernende den Lernprozess mit so ist dieser am eindringlichsten und zeigt die dauerhaftesten Ergebnisse. (vgl. Rogers, C.R., 1988, p. 175)

Rogers nennt unter anderem folgende Faktoren die vom Lernenden ausgehen, welcher als selbstbestimmt in seinen Möglichkeiten zu lernen betrachtet wird, welche signifikantes Lernen ermöglichen:.

- Der Lernende zeigt persönliches Engagement
- Der Lernende ist mit seinen Gefühlen und kognitiven Aspekten involviert.
- Der Lernende erkennt, ob das Gelernte seinen Bedürfnissen entspricht. (vgl. Rogers; Carl Ransom, 1969, p. 13)

Bis dato sind es die Schülerinnen und Schüler nicht gewohnt allzu viel Freiheit im Unterricht zu haben, daher muss darauf geachtet werden, dass die Bereitstellung von mehr Freiheit nicht dazu führt, dass sie sich im Chaos verlieren. (vgl. Motschnig-Pitrik & Derntl, Michael, 2002)

Für den personenzentrierten Unterricht eignen sich insbesondere Team- und Gruppenarbeiten, bei denen der Lehrende als Unterstützer und Berater fungiert. Durch die Arbeit in Gruppen werden bei den Lernenden mehrere Vorgänge gefördert:

- Lernende arbeiten und denken weitgehend selbstständig
- äußern ihre eigenen Gedanken
- lernen in Gruppen zu diskutieren
- lernen die Meinungen anderer kennen
- erarbeiten sich relativ selbstständig Erkenntnisse unter Verwendung der Hilfsmittel und Beiträge anderer
- haben die Möglichkeit zu produktivem und kombinierendem Denken
- verbessern ihre Leistungen aufgrund von Anregungen, Diskussionen und Erklärungen anderer in der Gruppe
- arbeiten auch bei zeitweiser Abwesenheit der Lehrperson. (vgl. Tausch, Reinhard & Tausch, Anne-Marie, 1991, p. 253 ff.)

Umsetzung mittels „In-ClassFlip“

Die Ideen des personenzentrierten Unterrichts passen gut zu dem Konzept des Flipped Classroom.

Das Unterrichtskonzept, welches im Rahmen dieser Arbeit erstellt wurde, basiert auf einer leichten Abwandlung des Flipped Classroom, da es im Informatikunterricht eher unüblich ist Hausübungen zu vergeben, werden beim In-ClassFlip auch die Inhalte im Unterricht erarbeitet und die Lehrperson steht auch in dieser Phase als Coach zur Verfügung.

Im Anschluss wird der Flipped Classroom beschrieben und im weiteren Verlauf auf die Umsetzung dieser Methode in Bezug auf das Unterrichtskonzept im Anhang eingegangen.

Flipped Classromm

*"The Flipped Classroom is an intentional shift of content
which in turn helps move students
back to the center of learning
rather than the products of schooling"
(Bennett, Brian E. et al., 2013)*

Das umgedrehte Klassenzimmer hat in den letzten Jahren als Antwort auf den Ruf nach technologieverbesserten, schülerzentrierten Lernumgebungen viel Aufmerksamkeit erhalten. Das Ziel dieser Methode ist, die wechselseitig beeinflussende Zusammenarbeit mit den Lehrenden auszuweiten und die klassischen Phasen des Erklärens ins Selbststudium zu verlagern. (vgl. Kim, M.K., Kim, S.M., Khera, O., & Getman, J., 2014)

J. Wesley Baker präsentierte dieses Konzept bei der „11th International Conference on College Teaching and Learning“, in Florida im Jahr 2000. Als Pioniere des „Flipped Classrooms“ werden meist Jonathan Bergmann und Aaron Sams genannt. In ihrem Buch „Flip Your Classroom“ (Bergmann ; Jonathan & Sams; Aaron, 2012) berichten sie über ihre Anfänge mit dieser Methode im Jahr 2006. Im Jahr 2012 gründeten sie dann gemeinsam mit April Gudenrath, Kristin Daniels, Troy Cockrum, Brian Bennett und anderen das gemeinnützige Flipped Learning Network (FLN). Ursprünglich diene das FLN um die Idee des umgedrehten Klassenzimmers zu verbreiten, heute steht das Teilen von Ressourcen, Tipps und Tools im Vordergrund. (vgl. „Who We Are,“)

Der traditionelle Unterricht ist stark lehrerzentriert. Die Lehrperson präsentiert Inhalte, gibt Erklärungen, während die Lernenden mit zuhören und mitschreiben beschäftigt sind. Die Geschwindigkeit ist für alle Schülerinnen und Schüler gleich, unabhängig von deren jeweiligen Grundvoraussetzungen, sowohl inhaltlich als auch motivational. Es bleibt wenig Raum für Dialog,

Fragen, Gruppenarbeiten, Diskussionen etc. Das Lösen von Aufgaben und das Üben werden oft fast zur Gänze nach Hause verlagert.

Beim „Flipped Classroom“ hingegen werden die Vorträge und Erklärungen in die Vorbereitungszeit der Schülerinnen und Schüler verlagert. Die Lehrkraft stellt für das Selbststudium Unterlagen bereit und die Lernenden können sich so mit den Inhalten im individuellen Tempo an einem Ort ihrer Wahl und so oft wie gewünscht bzw. nötig auseinandersetzen. Auf diese Weise werden die weniger hochwertigen kognitiven Prozesse wie das Verstehen und Erinnern (siehe Lernzieltaxonomie) aus dem Unterricht genommen und es bleibt im Unterricht Zeit für die oberen Stufen der Lernzielhierarchie. Laut dem Schema zur Klassifikation von Lehrzielen nach Bloom werden die Lernziele nach ihrer Komplexität gereiht und jede dieser Stufen baut auf die Stufe davor auf. Beim „Flipped Classroom“ werden die beiden unteren Ebenen in die Selbstverantwortung der Schülerinnen und Schüler gelegt und durch Materialien, welche von der Lehrperson zur Verfügung gestellt werden, unterstützt.

Da Tabletts und Smartphones zu den ständigen Begleitern der Schülerinnen und Schüler von heute zählen, bieten sich Erklärvideos bzw. Lernpfade zum Selbststudium an. Videos können von den Lehrkräften mit relativ wenig Aufwand selbst produziert werden (siehe Unterrichtskonzept: Tipps zur Videoerstellung) und gehören zur Lebenswelt der Lernenden. Sie eignen sich hervorragend zur Darstellung spezifischer Prozesse und Inhalte. Die Bilder dienen der Visualisierung und der Text zur Erklärung. Beides in Kombination führt zu mehr dualer Kodierung. Laut A. Paivio werden verbale und visuelle Informationen in unterschiedlichen Hirnarealen und auf verschiedene Art verarbeitet. So entstehen zwei unterschiedliche mentale Repräsentationen, welche jedoch inhaltlich durch eine referentielle Verknüpfung aufeinander Bezug nehmen. (vgl. Paivio, 1986)

Erste Erklärvideos wurden bereits in den 80er Jahren von Donald E. Knuth produziert, um Inhalte der Informatik zu erklären („Donald E. Knuth Lectures,” 2014)

Videos haben im Vergleich zu Büchern oder Skripten einige Vorteile. Sie sind fokussierter, zielgruppenorientierter, informeller und vor allem dynamischer. Es ist möglich Prozesse darzustellen und so den Lernenden die Bezüge zwischen den einzelnen Schritten klarzumachen bzw. den erforderlichen Denkprozess vorzuführen. Einige Tipps zur Erstellung von Erklärvideos befinden sich im Unterrichtskonzept. (siehe „Videos Erstellen“ im Unterrichtskonzept)

Der Betrachter erfährt die Leidenschaft des Präsentierenden und kann dort zur Langsamkeit geführt werden, wo sonst darüber gelesen werden würde. Zum Beispiel fällt es Lernenden oft schwer praktische Beispiele in Büchern nachzuvollziehen, da im Buch lediglich das Resultat, nicht aber der Prozess abgebildet werden kann und die Inhalte erscheinen meist steif, abweisend, starr und komplex.

Auch erfordert ein Buch mehr Verarbeitungsleistung, wodurch der Germane cognitive load erhöht wird. (siehe Bezug zu den Lerntheorien - Cognitive load Theorie)

Demzufolge bedeutet eine gute Mediengestaltung, dass man versucht, kognitive Überbelastung zu vermeiden.

Das heißt wiederum, dass das bloße Ansehen der Videos zu wenig ist, um als lernförderliche Aktivität gesehen werden zu können. Ohne Aufwand stellt sich kein Lernerfolg ein. Daher sollten, begleitend zu den Videos, unbedingt schriftliche Lernmaterialien zur tieferen und aktiven Beschäftigung mit den Inhalten bereitgestellt werden.

Alle Lernunterlagen können den Schülerinnen und Schülern digital zur Verfügung gestellt werden oder aber auch, mit Ausnahme der Videos, in Papierform.

Herr Prof. Christian Spannagel spricht in einem seiner Vorträge von den 10 Irrtümern beim Umgang mit digitalen Medien im Unterricht:

- *„Die Schülerinnen und Schüler können schon alles.“* (sind Digitalnatives) Es wäre falsch anzunehmen, dass alle Schülerinnen und Schüler versiert im Umgang mit Technologie sind.
- *„Die Schule ist nicht zuständig.“* Es ist jedoch Aufgabe der Schule, Schülerinnen und Schüler zu mündigen, selbstständigen Bürgern zu erziehen.
- *„Die Schule muss überwiegend vor den Gefahren warnen.“* Hervorgehoben werden sollten aber vor allem die nützlichen Aspekte der Technologie. In diesen Kontexten kann man dann besprechen, was zu beachten ist.
- *„Computer lösen Lehrer ab.“* Nein! Computer unterstützen und können genutzt werden, um das Lernen zu optimieren.
- *„Digitale Medien machen lernen leichter.“* Der Lernprozess wird dadurch nicht leichter und erfordert Kapazitäten für höhere kognitive Prozesse. Lernen und üben müssen die Schülerinnen und Schüler trotzdem selbst.
- *„Digitale Medien lösen analoge ab.“* Nein! Sie erhöhen lediglich die Vielfalt.
- *„Ich muss mich mit diesen Technologien nicht auseinandersetzen.“* Falsch. Es ist Aufgabe der Schule den Schülerinnen und Schülern die Verwendung der neuen Technologien in diesem Zusammenhang zu vermitteln.
- *„Mit der nachrückenden Generation ändert sich alles.“* Nein. Die Möglichkeiten Inhalte zu vermitteln werden deutlich umfangreicher.
- *„Zeit, die ich jetzt in digitale Medien investiere, spare ich mir später.“* Dafür ist die Entwicklung von neuen Technologien zu schnell. Weiterentwicklung der Lehrqualität ist zeitintensiv.

- „*Ich kann das nicht.*“ Es braucht Mut zu Fehlern. Die Lehrperson wird selbst zum Lernenden. (Spannagel; Christian)

Materialien können mit offenen Lizenzen wie Creative Commons, GNU General Public License oder Open Publication License (OPL) über das Internet öffentlich gemacht werden. (vgl. „Open Education,“ 2016) Die Creative Commons helfen Nutzungsrechte auf einfache Art kostenlos zu regeln und sollten dazu beitragen, dass möglichst viele Urheber ihre Werke der Öffentlichkeit zugänglich machen.

Unter „<http://creativecommons.org/license>“ kann man sich über die passende Lizenz informieren und diese dann mithilfe eines Generators erstellen.

In puncto „öffentlich machen“ übernimmt im deutschsprachigen Raum Prof. Christian Spannagel von der Pädagogischen Hochschule Heidelberg eine Vorreiterrolle. Er stellt alle seine Vorlesungen, Kurse und Ideen ohne Zugangsbeschränkungen ins Internet. Zum Beispiel findet man seine Vorlesungen auf YouTube und seine Stundenvorbereitungen und Vorträge über seine Home Page. In einem Vortrag zu Open Education erläutert er seine Beweggründe: (vgl. Christian Spannagel, 2013)

- „*Murmeltiervermeidung*“: keine Langeweile für den Vortragenden (semesterweise grüßt das Murmeltier) und mehr Raum für Diskussionen mit den Studierenden
- „*Kleiderschrankfrage*“: aus einer Vielzahl von bereitgestellten Unterlagen kann das passende für den Tag gewählt beziehungsweise arrangiert werden
- „*Gehirn-Extension*“: Bei der Erstellung der Unterlagen oder Videos für die Öffentlichkeit ist man gefordert mehr auf Qualität zu achten. Außerdem kann man sich später Inhalte von sich selbst erklären lassen.
- „*Josef Hutzl*“: Die Geschichte eines Fernfahrers, welcher mit den YouTube-Videos Mathematik lernte und später sogar ein Verfahren zum Quadrieren entwickelte.
- „*Prof. Brenner*“: Ein Professor der auf Fehler in einer Vorlesungsaufzeichnung hinwies. (auch Fehler sollten öffentlich diskutiert werden, denn daraus resultiert ein konstruktiver Umgang und eine verbesserte Fehlerkultur)
- „*Denkdruck & rauher Wind*“: Das Schreiben als Denkwerkzeug. Man ist gezwungen seine Gedanken in sinnvolle Worte zu fassen und Diskussionen, sowie Kritik helfen beim Nachdenken und Reflektieren.
- „*Crowd Thinking*“: Brainstorming im Internet. Mitmachakquise: Aufforderung zur Beisteuerung von Forschungsideen und zur Mitarbeit bei wissenschaftlichen Papers
- „*Being Famous*“: Leistungen werden durch die Breitenwirkung des Webs mehr gewürdigt.
- „*Stimulus-Garantie*“: Informationsverarbeitung als Grundbedürfnis
- „*Kostenreduktion*“: weniger Druckkosten

Prof. Christian Spannagel sieht sich selbst als Organisator kollektiver Wissenskonstruktion.

Bei oberflächlicher Betrachtung der „Flipped Classroom“ Methode könnte man annehmen, dass die Vermittlung der Stoffinhalte und damit die Erstellung der Lernunterlagen das Kernstück ist. Diese dienen jedoch nur zur Vorbereitung auf den eigentlichen Unterricht, in dem (da ja bereits alle Schülerinnen und Schüler mit den Inhalten vertraut sind) nun endlich was Vernünftiges gemacht werden kann, nämlich „Denken“. Man kommt zusammen, um etwas zu tun: für vertiefende Aufgaben, Diskussion von Problemen, Klären von Fragen, persönliche Betreuung, tiefere Diagnose etc. (vgl. Christian Spannagel, 2013)

Der Zweck dieser Methode ist es, mehr Zeit zur Interaktion zwischen Lernenden und Lehrenden sowie für die Anwendung von Wissen zur Verfügung zu haben.

Die Präsenzphase soll helfen, das Verständnis der Schülerinnen und Schüler zu fördern und ihnen helfen, das Gelernte zu verarbeiten und in Bezug zu ihrer Lebenswelt zu setzen. (vgl. Spencer, Dan, Wolf, Deb, & Sams, Aaron, 2011)

Die Rolle der Lehrperson verändert sich. Aus dem Dozenten wird ein Lernbegleiter und somit wird seine Funktion noch wichtiger. Es bedarf einer anderen Haltung beziehungsweise Einstellung der Lehrenden und sie müssen versuchen aus Lernendenperspektive zu denken und die Lernumgebung so zu gestalten, dass man weg vom Stoff und hin zum Verständnis kommen kann. (vgl. Kent & Noss; 2001)

Der Lehrende ist nicht mehr in der Expertenrolle, sondern Lerndesigner und Coach, ganz im Sinne des personenzentrierten Unterrichts. (siehe Personenzentrierter Unterricht)

Zu beantworten ist die Frage nach den Lernzielen und wie diese jeder auf seine ganz persönliche Art erreichen kann. Die Lehrperson ist bei diesem Konzept näher an den Lernenden und kann somit besser einschätzen, wo die Probleme und Schwierigkeiten liegen und daher besser und individueller unterstützen.

Die weiteren Methoden im Unterricht sind natürlich abhängig vom Inhalt, hier ist die Lehrperson gefordert eine konstruktive Lernumgebung zu gestalten. Besonders geeignet erscheinen Varianten des kooperativen Lernens wie:

- **Team- und Gruppenarbeit** zum Bearbeiten von vertiefenden Beispielen und Transferaufgaben
- **Think-Pair-Share:** Hierbei handelt es sich um eine kollaborative Unterrichtsstrategie die 1981 von Frank Lyman von der Universität in Maryland entwickelt wurde. Erst denkt jeder Lernende über eine Frage eigenständig nach (Think) anschließend werden die Ergebnisse mit einem Partner verglichen, diskutiert und abgestimmt (Pair) und erst dann werden die Antworten mit der ganzen Klassen geteilt und diskutiert (Share) (vgl. „Think-pair-share,“ 2017)

- **Aktives Plenum:** Hier leitet eine Lernende oder ein Lernender die Diskussion, wodurch die Vermittlerrolle der Lehrperson verhindert wird. Dieser notiert Anregungen aus dem Plenum und stellt Rückfragen, falls erforderlich. (vgl. Lutz Berger, Joachim Grzega, Christian Spannagel, 2009)
- **Pro- und Contradebatte:** Hier teilt man die Klasse in zwei Hälften, bestimmt einen Moderator und einen Protokollanten. Eine Hälfte argumentiert Pro, wogegen die andere Hälfte nach Gegenargumenten sucht. Diese Methode ist strenger geregelt als eine Diskussion und sollte als ein zeitlich befristetes Streitgespräch auf gehobenem Niveau verstanden werden. (vgl. Nicole Sürig, 2006)
- **Dialog:** Ein Gespräch in einer stressfreien Atmosphäre, zwischen zwei oder mehr, zum offenen Austausch bereiter Personen, welches durch gemeinsames Denken intensiviert und vertieft wird. Damit bietet sich die Möglichkeit Verhaltensmuster zu durchbrechen und besser voran zu kommen, sowohl in der Beziehung zu anderen als auch zur Umwelt. Um einen Dialog zu ermöglichen bedarf es einer dem personenzentrierten Ansatz entsprechenden Haltung der Beteiligten. Wesentlich ist vor allem, dass aufmerksam und aktiv zugehört wird. Der Dialog beschäftigt sich meist mit einem gemeinsamen Anliegen und verbindet Teilansichten zu einem Ganzen. Im Gegensatz zur Diskussion bei welcher der Fokus vorwiegend auf den Unterschieden in den Einzelmeinungen liegt, richtet sich hier der Blick auf die Verbindung zwischen den Teilen. Es ist ein ehrlicherer Gedankenaustausch, in welchem sich jede Person einbringen kann und ernstgenommen wird, aber auch die Beziehungen zwischen den Personen und den Themen Platz finden. Die Basis für Dialogfähigkeit stellen nach William Isaac (vgl. *Isaacs, W.*, 1999; p. 164) zitiert in: (Motschnig Renate & Nykl Ladislav, 2013, p. 195) „Zuhören“, „Respekt“, „Suspendieren“ und „Artikulieren“. Suspendieren bedeutet, dass die eigenen Gedanken beiseite gelegt werden, um offen für die andere Person und deren Aussagen zu sein. So kann es möglich werden Verschiedenheiten zu schätzen, aus diesen zu lernen, destruktive Missverständnisse zu überwinden und die persönliche Sichtweise zu erweitern, gemeinsam zu denken, sowie eine gemeinsame Bedeutung zu schaffen. Die Dialogpartner können neue Erkenntnisse erlangen, inspiriert werden, aus ihrem eigenen Denkraum befreit werden und zwischenmenschliche Beziehungen verbessern. (vgl. Motschnig Renate & Nykl Ladislav, 2013, Chapter 10)

Beim kooperativen Lernen sind die Lernenden gezwungen ihre Gedanken kohärent zu strukturieren und sprachlich zu äußern (es kommt zur Reibung mit den Inhalten), soziale Kompetenzen werden gefördert und individuelle Wissenslücken werden deutlich.

InClassFlip im Unterrichtskonzept dieser Arbeit

Im Sinne des „Flipped Classrooms“ und des personenzentrierten Unterrichts wurden die Inhalte zur „Einführung in die Programmierung mit dem App Inventor“ so aufbereitet, dass es den Lernenden weitestgehend möglich sein sollte, sich diese selbstständig zu erarbeiten.

Es wurden Arbeitsblätter gestaltet, Internetlinks und Videos zur Verfügung gestellt.

Ergänzend dazu sind Teamarbeiten, Gesprächsrunden, Präsentationen, Feedbacks und Reaktionsblätter geplant, um signifikantes Lernen zu ermöglichen und die Kommunikationsfähigkeiten der Schülerinnen und Schüler zu verbessern.

Online Materialien

*„Online-Bereitstellung von geistigem Material
lässt mehr Raum
für eine sinnvolle Kommunikation im Unterricht“.
(Motschnig-Pitrik, Renate & Derntl, Michael, 2008)*

In der Erarbeitungsphase können die Lernenden selbst entscheiden in welchem Tempo sie vorangehen, was besonders den schwächeren Schülerinnen und Schülern zugute kommt. Da die Materialien auch außerhalb der Unterrichtszeit zur Verfügung stehen, haben interessierte Lernende die Möglichkeit, zeit- und ortsunabhängig, die Inhalte weiter zu vertiefen.

Die Materialien können von der Lehrperson leicht hochgeladen und an die Bedürfnisse der Gruppe angepasst werden. Auch die Lernenden können zu deren Verbesserung beitragen, indem sie Material suchen und dieses für die gesamte Gruppe verfügbar machen oder vorhandenes Material mit Kommentaren zu Inhalt, Qualität und / oder Verfügbarkeit versehen.

Insbesondere im Informatikunterricht ist es sinnvoll das Internet als Wissensquelle für exploratives, offenes Lernen zu nutzen (vgl. Cornelius-White; J., 2007) und es ist besonders für den personenzentrierten Ansatz geeignet, da die Studierenden das semantische Web erforschen und bei der Erfüllung ihrer Ziele kooperieren können. (vgl. Motschnig-Pitrik, Renate & Derntl, Michael, 2008)

Zusätzlich lernen sie Inhalte selbstständig und selbstverantwortlich zu erarbeiten und legen damit einen guten Grundstein für lebenslanges Lernen. Die Bedeutsamkeit ihrer Arbeit steigt, da sie mit ihrem Wissen während der Präsenzphase einen Beitrag zur Gemeinschaft in der Klasse leisten.

Es müssen lediglich einige minimale technologische Anforderungen erfüllt werden. Zum Beispiel genügen ein E-Mail Account für Lernende und Lehrende, sowie ein öffentlicher Webspaces zum Hochladen und Hosten von Dateien. Diese minimalen Mittel ermöglichen bereits die Umsetzung, aber

für eine differenziertere und komfortablere Handhabung von Onlineaktionen und -Interaktionen werden integrierte virtuelle Lernumgebungen empfohlen.

Werden auch die von den Lernenden erarbeiteten Projekte auf der gemeinsam genutzten Plattform für alle öffentlich zugänglich gemacht und gegebenenfalls mit Kommentaren des Lehrenden versehen können die Lernenden aus einer Vielzahl an Projekten lernen und nicht nur anhand ihres eigenen. (vgl. Motschnig-Pitrik, Renate & Derntl, Michael, 2008)

Teamarbeit im Projekt, Gruppendiskussionen, Präsentationen und Feedback

Wie bereits erwähnt eignen sich Gruppen- und Teamarbeiten besonders für den personenzentrierten Unterricht (vgl. Tausch, Reinhard & Tausch, Anne-Marie, 1991, p. 253) (vgl. Barrett-Lennard, Godfrey T, 1998, p. 183).

Projektbasierte Lernszenarien eignen sich besonders, um signifikantes Lernen zu ermöglichen, da sie realen Situationen ähneln.

Auch erfordert der personenzentrierte Ansatz eine intensive Kommunikation zwischen allen Beteiligten, diesem Erfordernis wird mit den verschiedenen Präsentationen und Gruppendiskussionen Rechnung getragen. Auf diese Weise können offene Fragen gemeinsam geklärt, Ergebnisse aus der Projektarbeit gesichert und geteilt werden. Die Schülerinnen und Schüler haben so die Möglichkeit von mehreren Beispielen zu lernen und nicht nur von einem einzigen. Es können auf diese Weise viele Aspekte angesprochen und geklärt werden, denn was der Eine vielleicht nicht sagt, wird möglicherweise von einem Anderen angesprochen.

Darüber hinaus sollen die Lernenden ein offenes und konstruktives Feedback geben und dabei auch erfahren wie ihre Rückmeldung von anderen wahrgenommen wird. Ein wichtiger Aspekt dabei ist auch das Annehmen von Kritik und der Umgang damit.

Kritische Reflexion und das Bewusstmachen für die eigenen Stärken und Schwächen erfordert Verantwortung zu übernehmen und ist daher einer der Schlüssel zur persönlichen Verbesserung und zum selbst initiierten Lernen. (vgl. Rogers; Carl Ransom, 1969, p. 158 ff.)

Reaktionsblätter

Bieten die Möglichkeit jegliche Art von Reflexionen, Reaktionen und Kommentaren der Schülerinnen und Schüler zu sammeln, welche diese in Bezug auf einzelne Einheiten oder das gesamte Modul kundgeben möchten. Im Vordergrund steht dabei, dass die Lernenden dazu motiviert werden den Unterricht bewusst zu reflektieren, innere Bedeutungen zu finden und den Lehrenden bei der weiteren Gestaltung der Einheiten, entsprechend den Bedürfnissen der Lernenden, zu unterstützen.

Reaktionsblätter sind ein einfaches und effektive Mittel, um Schülerinnen und Schüler als ganze Personen auf allen Ebenen des Lernens wahrzunehmen und als Hilfe zur Verbesserung der Qualität von Kursen unter der Voraussetzung, dass es sich um personenzentrierten Unterricht handelt. Die eigentliche Herausforderung liegt im zwischenmenschlichen Umgang mit den Reaktionen und dem Agieren auf die geäußerten Gefühle, Bedeutungen, Beobachtungen und Suggestionen der Schülerinnen und Schüler. Denn die Reaktionsblätter dürfen nicht einfach nur gesammelt werden, sondern sollen auch in der darauffolgenden Einheit mit der ganzen Gruppe besprochen werden. (vgl. Motschnig-Pitrik, Renate & Derntl, Michael, 2008)

Bezug zu den Lerntheorien

Konstruktionismus

Seymour Papert entwickelte den Konstruktionismus. Ein lerntheoretischer Ansatz und eine Strategie für Bildung. Dieser basiert auf zwei Arten von Konstruktion:

- Basierend auf den Theorien von Jean Piaget ist Lernen ein aktiver Prozess bei welchem Menschen aktiv Wissen aus ihren Erfahrungen in der Welt konstruieren.
- Lernende erwerben neues Wissen besonders effektiv, wenn sie sich mit der Konstruktion persönlich bedeutungsvollen Elementen beschäftigt. Wichtig ist dabei, dass sie etwas schaffen, was für sie selbst oder für andere sinnvoll ist, indem sie aktiv daran arbeiten. (vgl. Resnick; Michael, 1996)

Beeinflusst wurde Papert von Jean Paul Piaget, einem Vertreter der Konstruktivismus, der vom Lernen als „Aufbau von Wissensstrukturen“ spricht, welcher unabhängig von den Umständen ist, in denen das Lernen stattfindet. Diese Theorie ergänzte er mit der Idee, dass Lernen besonders gut gelingt, wenn der Lernende bewusst mit der Konstruktion eines Werkes beschäftigt ist. Weiters wird die Idee vertreten, dass manche Menschen Denkweisen bevorzugen, die physischen Objekten nahe sind während andere abstrakte und formale Mittel verwenden, um sich von konkreten Objekten zu distanzieren. Beide Ansätze sind für den Konstruktionismus relevant.

Mit LOGO (siehe visuelle Programmierung) hat Papert etwas mitentwickelt, um die Lern- und Denkweise von Kindern zu verbessern. Computernetzwerke sollen nicht nur als Medium für die Verbreitung von Informationen betrachtet werden, sondern in erster Linie als neues Medium für die Konstruktion von Wissen. (vgl. Papert Seymour & Harel, Idit, 1991)

Eine Erweiterung dieser Theorie ist der verteilte Konstruktionismus, welcher sich speziell auf Situationen bezieht, in denen mehr als eine Person an den Aktivitäten, unter Verwendung von Computernetzwerken, beteiligt ist und berücksichtigt, dass Kognition und Intelligenz aus der Interaktionen eines Menschen mit anderen und der Umwelt entstehen.

Es wird behauptet, dass kollaborative Aktivitäten besonders effektiv sind um Wissen zu bilden. Drei der wichtigsten Aktivitäten in diesem Sinn sind:

Diskussionen

Dabei sind nicht nur Gruppendiskussionen, sondern auch der Austausch zu Konstruktionen in elektronischer Form, wie zum Beispiel E-Mails, Newsgroups etc., in denen Schülerinnen und Schüler Tipps, Ideen und Strategien teilen können gemeint. Diese Diskussionen erleichtern und inspirieren die Arbeit der Lernenden und diese kommen somit besser voran und schaffen qualitativ hochwertigere Produkte.

Teilen von Werken

Dadurch können die Schülerinnen und Schüler Werke anderer ausprobieren und Teile davon kopieren und wiederverwenden. Sie können Dinge schaffen, welche andere sehen und daraus einen Nutzen ziehen. Besonders computerunterstützte Lernumgebungen und das Internet erleichtern das Teilen von Produkten.

Zusammenarbeit

Auch die Kooperation zwischen den Lernenden wird durch Computernetzwerke wesentlich erleichtert. Lernende sind dadurch in der Lage, nicht nur zu teilen, sondern auch in Echtzeit an Projekten zusammenzuarbeiten.

Computernetzwerke und das Internet verändern mit dieser Art von Aktivitäten nicht nur den Lernprozess, sondern auch den Inhalt dessen was gelernt wird. Sie erleichtern kollaborative Projekte und die Bereitstellung der benötigten Materialien für die Erforschung von Phänomenen. (vgl. Resnick; Michael, 1996)

Cognitive Load Theorie

Die cognitive load Theorie (Chandler & Sweller, 1992) basiert auf der Annahme, dass die Kapazität des Arbeitsgedächtnisses begrenzt ist und unterscheiden drei Formen des cognitive load

- **Intrinsic cognitive load:** Ist bedingt durch die Schwierigkeit und Komplexität des Inhalts und kann nicht durch Gestaltung beeinflusst werden, sollte aber möglichst gering gehalten werden.
- **Extraneous cognitive load:** Entsteht durch instruktionale Bedingungen und kann gut durch gestalterische Mittel reduziert werden.
- **Germane cognitive load:** Ist die vertiefende Auseinandersetzung mit den Inhalten und daher wünschenswert. (vgl. Jong de; Ton, 2009)

Die Grundlage dieser Theorie bilden folgende Annahmen:

- Für Problemlösungs- und Verarbeitungsprozesse ist das Arbeitsgedächtnis zuständig.
- Es kann nur eine begrenzte Anzahl an Elementen gleichzeitig im Arbeitsgedächtnis gehalten werden. (vgl. Miller G. A., 1956) und es ist zeitlich begrenzt.
- Im Arbeitsgedächtnis können nur zwei oder drei Informationen gleichzeitig behandelt werden.
- Das Arbeitsgedächtnis teilt sich in einen visuellen und einen verbalen Bereich. (vgl. Duale Codierung nach A. Pavio bzw. Split-Attention-Effekt nach Mayer, 1998)
- Der Mensch ist sich des Langzeitgedächtnisses nicht unmittelbar bewusst.

Das Langzeitgedächtnis hat eine virtuell unbegrenzte Kapazität. Seine Ressourcen werden vom Arbeitsgedächtnis, welches ja nur über begrenzte Kapazität verfügt zur Erweiterung genutzt. (vgl. Sweller, John, Merrienboer van; Jeroen J.G., & Paas, Fred G.W.C., 1998)

Im Langzeitgedächtnis wird Wissen in Form von Schemata gespeichert und organisiert. Liegen diese Schemata automatisiert vor, so können auch komplexere Zusammenhänge im Arbeitsgedächtnis verarbeitet werden, ohne dass auf freie kognitive Ressourcen zurückgegriffen werden muss. Denn ein Schema kann durchaus komplex sein, also alles was gelernt und als eine Einheit betrachtet wird, beinhalten. Unterelemente eines Schemas, wie komplex es auch immer sein mag, benötigen keine Arbeitsspeicherkapazität mehr. (ebd.)

Diese Theorie wurde vor allem dazu entwickelt, um Lehrende zu unterstützen Materialien so aufzubereiten, dass Schülerinnen und Schüler ihre optimale intellektuelle Leistung erreichen können.

Die Leichtigkeit, mit der Informationen im Arbeitsgedächtnis verarbeitet werden können, ist ein Hauptanliegen der kognitiven Lasttheorie. Die Belastung des Arbeitsgedächtnisses kann entweder durch die intrinsische Natur des Materials oder alternativ durch die Art und Weise, in der das Material präsentiert wird, oder durch die Aktivitäten vom Lernenden und die erforderlichen Tätigkeiten beeinflusst werden. Die intrinsische kognitive Belastung kann nicht durch instruktive Interventionen verändert werden. Jedoch ist die externe kognitive Belastung eine unnötige Belastung und kann durch Instruktionsinterventionen verändert werden. (ebd.)

Eine ähnliche Theorie vertritt Mayers Multimediaprinzipien (vgl. Mayer, Richard E.; 2009) (siehe Videos erstellen im Unterrichtskonzept – Design- und Gestaltempfehlungen)

Blooms Lernzieltaxonomie

Lernziele sind laut Robert F. Mager „[...] zweckmäßige Zielbeschreibungen [...], mit denen es gelingt, die Unterrichtsabsichten [...] mitzuteilen. (Mager, R.F., 1972, p. 19)

Man unterscheidet zwischen:

- **Kognitiven Lernzielen**
Betont werden die Erinnerung bzw. das Reproduzieren eines Stoffes, sowie das Kombinieren von Methoden und Verfahren.
- **Affektiven Lernzielen**
Hier stehen Gefühle, Einstellungen, Werte, Emotionen und Interessen in Vordergrund.
- **Psychomotorischen Lernzielen**
Hauptaugenmerk liegt auf motorischen, sensorischen oder physischen Fertigkeiten wie Schrift, Sprache, Bewegung, Handwerk und Technik.

Eine Lehrzieltaxonomie ist somit ein Schema zur Klassifikation von Lehrzielen.

Blooms Modell bezieht sich ausschließlich auf den kognitiven Teil der Lehrziele und soll helfen zu verstehen, wie Lernen geschieht. Es ist hierarchisch aufgebaut und unterscheidet sechs Stufen.

Jede dieser Stufen baut auf der Stufe davor auf. So werden Lernziele nach ihrer Komplexität gereiht. Beginnend mit simpler Wiedergabeleistung bis zu komplexen Problemlösungs- und Begründungsaktivitäten.

Das Modell dient der Ordnung von Lernzielen und soll helfen diese, trotz ihrer Vielfalt, nach logischen Kriterien zu strukturieren.

Indem Lernziele einer Kategorie zugeordnet werden, kann man deren Komplexität verstehen.

Dadurch sollen Lehrende angeregt werden, ihren Unterricht bewusster vorzubereiten und gestalten. Klare Lernziele, um den Schülerinnen und Schüler und ihren Bedürfnissen bezüglich des Lernens gerecht zu werden, können leichter definiert und gegebenenfalls mit anderen Lernzielkatalogen verglichen werden. Die Lehrzieltaxonomie ist daher hilfreich bei der Planung von Lernangeboten. Den Lehrenden werden damit Möglichkeiten aufgezeigt und es wird ihnen erleichtert, Erwartungen an eine Lernumgebung sinnvoll zu treffen.

Ebenfalls ist es möglich damit die Lernzielkontrolle zu erleichtern.

Die Taxonomie gilt für alle Fachgebiete und hat einen hohen Abstraktionsgrad.

Die einzelnen Stufen werden noch in Unterstufen geteilt. Daraus ergibt sich folgendes Schema. (vgl. Anderson, Lorin W. u. a., 2001)

Wissen

Hier steht das Erinnern im Vordergrund. Es handelt sich um ein unverändertes Abrufen des Gelernten.

- Kenntnis konkreter Einzelheiten:
 - Details: spezifisch und isoliert – sehr geringer Abstraktionsgrad
 - Begriffe: Bedeutung von Fachbegriffen, Symbolen
 - Fakten: Ereignisse, Personen, Orte und Zeiten
Beispiel: Grundlagenwissen über Organismen. Wissen einer Vielzahl von Begriffen und ihrer Bedeutung
- Kenntnis wie mit konkreten Einzelheiten umgegangen wird und wie diese behandelt werden können:
 - Regeln und Konventionen
Man kann sich erinnern, wie mit Mustern, Dingen umgegangen, wie diese organisiert und bewertet werden können.
Beispiel: Schülerinnen und Schüler können die Hauptarten literarischer Gattungen verstehen.
 - Trends und Zeitfolgen
Man kann sich an den zeitlichen Verlauf von Prozessen, Ereignissen, Bewegungen erinnern.
 - Einordnung und Kategorien
Wissen über Klassen, Gruppen, Gliederung von Ereignissen erinnern welche für ein bestimmtes Fachgebiet grundlegend sind.
 - Kriterien
Beispiel: Lernende können Unternehmungen beurteilen
 - Methodologisches Wissen
Beispiel: Methoden, die zur Lösung eines mathematischen Problems verwendet werden können.
- Kenntnis der Universalien und Abstraktionen eines Gebiets
 - Prinzipien und Verallgemeinerungen
Beispiel: Das Wissen über wichtige Verallgemeinerungen einer bestimmten Kultur
 - Theorien und Strukturen
Beispiel: Schülerinnen und Schüler können ziemlich vollständig die Theorie der Evolution wiedergeben.

Schlagworte: erkennen, erinnern, angeben, aufschreiben, darstellen, reproduzieren, wiedergeben (ebd.)

Verstehen

Schülerinnen und Schüler verstehen, was ihnen mitgeteilt wird und sind in der Lage dieses Material zu nutzen. Sie können jedoch nicht alle Zusammenhänge zu anderen Informationen erkennen.

- **Übersetzung**

Inhalte können mit eigenen Worten wiedergegeben werden oder in andere „Sprachen“ übersetzt werden. Es können auch Metaphern, Symbole, Ironie etc. erkannt werden.

Beispiel: Mathematisches Textbeispiel in Formeln übersetzen und umgekehrt.

- **Interpretation**

Informationen können erklärt oder zusammengefasst werden. Dabei werden diese neu strukturiert.

Beispiel: Die Fähigkeit Daten zu interpretieren.

- **Extrapolation**

Die Fertigkeit über gegebene Informationen hinaus Auswirkungen, Implikationen und dergleichen zu bestimmen.

Schlagworte: interpretieren, erläutern, erklären, zusammenfassen, ableiten, vergleichen, begründen, beschreiben, umschreiben, einordnen. (ebd.)

Anwenden

Der Gebrauch von Abstraktionen in einzelnen, konkreten Situationen. Die Schülerinnen und Schüler sind in der Lage, das Verstandene zur Lösung eines Problems einzusetzen.

Schlagworte: ausführen, umsetzen, berechnen, einordnen, erstellen, anknüpfen, aufstellen, realisieren, unterscheiden, umschreiben. (ebd.)

Analyse

Die Fähigkeit Nachrichten in ihre Bestandteile zu zerlegen um ihre Beziehungen und Ideen klar zu machen. Hierfür werden die Fertigkeiten der unteren Stufen benötigt.

- **Elementen-Analyse**

Inhaltliche Elemente können identifiziert werden. Man kann Hypothesen von Tatsachen unterscheiden.

- **Relationen-Analyse**

Die Beziehungen zwischen inhaltlichen Elementen kann verstanden und klar gemacht werden.

- **Analyse von organisatorischen Prinzipien**

Explizite und implizite Strukturen, welche Nachrichten zusammenhalten, erkennen und verstehen.

Schlagworte: differenzieren, auflösen, gliedern, klassifizieren, untersuchen, organisieren, zuordnen. (ebd.)

Synthese

- Produktion einer einzigartigen Kommunikation

Die Fertigkeit persönliche Erfahrungen, Ideen und Gefühle schriftlich oder mündlich mitzuteilen.

- Produktion eines (Handlungs-) Plans

Die Fähigkeit Arbeitspläne als Weg zur Erfüllung von Aufgaben zu erstellen.

- Ableitung einer Sammlung abstrakter Beziehungen

Man kann auf Grundlage der Analyse geeignete Hypothesen aufstellen und diese gegebenenfalls adaptieren.

Schlagworte: überprüfen, kritisch betrachten, auswerten, beurteilen, widerlegen, qualifizieren, vereinfachen, folgern, bewerten. (ebd.)

Evaluation

- Beurteilung auf Grundlage interner Evidenz

Inhalte können nach innerer Stimmigkeit, logischer Entsprechung beurteilt werden.

- Beurteilung auf Grundlage externer Kriterien

Inhalte können nach Kriterien anderer Zusammenhänge beurteilt werden. (ebd.)

Visuelle Programmierung

*„An algorithm must be seen
to be believed,
and the best way to learn
what an algorithm is all about is to try it“.
(Knuth; Donald E., 1997, p. 4)*

Die Art der Softwareentwicklung, bei welcher hauptsächlich graphische Notationen und interaktiv manipulierbare Komponenten zum Einsatz kommen, wird nach Stephan Schiffer als visuelle Programmierung verstanden. (vgl. Schiffer, Stephan, 1998)

Die Anfänge visueller Programmierung liegen weit zurück. Seymour Papert entwickelte zum Beispiel bereits 1967, mit Bolt, Beranek and Newman bereits, die Programmiersprache LOGO am Massachusetts Institute of Technology. (vgl. "Logo History,") Mit dieser verband Papert auch seinen Lerntheoretischen Ansatz des Konstruktivismus (siehe Bezug zu den Lerntheorien - Konstruktivismus).

Seither hat die Entwicklung visueller Programmiersprachen und visueller Programmierumgebungen, motiviert dadurch, den Einstieg in die Programmierung zu erleichtern, stark zugenommen und es haben sich unterschiedliche Formen entwickelt.

Paradigmen der visuellen Programmierung

Es gibt verschiedene Paradigmen der visuellen Programmierung: (vgl. Schiffer, Stephan, 1998, p. 98 ff.)

- **Kontroll- bzw. Steuerflussorientierte Systeme:** Diese entsprechen der imperativen Programmierung und können mit Modellen wie Petri-Netzen, erweiterter EPK, Business Process Model and Notation (BPMN) dargestellt werden.
- **Objektorientierte Systeme:** Objekte mit öffentlichen Eigenschaften kommunizieren in diesen Systemen untereinander, sind Instanzen einer Klasse und Vererbung wird unterstützt. (vgl. Kay, Alan C.; 1993). Sie werden zum Beispiel mit Aktivitätsdiagrammen, Use-Case-Diagrammen und objektorientierter EPK modelliert.
- **Datenflussorientierte Systeme:** Die Reihenfolge der Ausführung von Befehlen hängt von der Verfügbarkeit der Daten ab. Zur Modellierung dieser Systeme sind Petri-Netze und Datenflussdiagramme geeignet.
- **Funktionsorientierte Systeme:** Für die automatisierte Bearbeitung mathematischer Funktionen. Beispiele hierfür sind Wettervorhersageberechnungen und Simulationssysteme. (vgl. Dumke, Rainer; 2013, p. 259)

- **Regelorientierte Systeme:** Eine Folge von Bildtransformationen mit optionalen Bedingungen und Aktionen legt bei diesen Systemen den Programmablauf fest. Beispiele hierfür sind LEGOsheet (Gindling Jim, et al, 1995) und StageCast von Allen Cypher und David Canfield Smith ("Stagecast Creator,")
- **Constraint-orientierte Systeme:** Bestehen aus Objekten und ständig geltenden Beziehungen zwischen diesen Objekten (Constraints) und entsprechen konzeptionell der logischen Programmierung .(vgl. Schiffer, Stephan, 1998, p. 117 ff.) Beispiele hierfür sind der GameSalad Creator ("GameSalad – Empowering Game Creators,") und Algodoo ("Algodoo,")
- **Beispielorientierte Systeme:** Das System zeichnet Benutzeraktionen auf und wiederholt diese später.
- **Formularorientierte Systeme:** Beruhen auf den Konzepten der Tabellenkalkulation wie zum Beispiel Excel.
- **Multiparadigmenorientierte Systeme:** Stellen eine Kombination aus mehreren der o.a. Systemen dar.

Die Anzahl der Paradigmen visueller Programmierung zeigt, dass viele unterschiedliche Ansätze bereits verfolgt wurden. Den vermutlich größten Erfolg erzielten dabei die sogenannten blockbasierten Sprachen, welche gerne verwendet werden, um Programmieranfängern den Einstieg in die Programmierung zu erleichtern.

Diese, im weiteren ausführlicher diskutierte, blockbasierte visuelle Programmierung folgt in den meisten Fällen, vor allem dem Steuerfluss- bzw. Kontrollflussparadigma sowie auch dem Paradigma der Objektorientierung.

Blockbasierte Programmierumgebungen

Blockbasierte Programmierumgebungen ermöglichen es den Anwenderinnen und Anwendern Programme zu erstellen, indem sie Code-Blöcke zusammensetzen. Die einzelnen Blöcke repräsentieren Programmcode, haben spezielle Formen und Farben und können meist per Drag-and-Drop zusammengefügt werden. Außerdem stehen bereits viele vorgefertigte Prozeduren mit sprechenden Bezeichnungen zur Verfügung.

Diese Programmierumgebungen wurden in erster Linie dazu entwickelt, um Einsteigern das Coding zu erleichtern und werden bereits vermehrt im Unterricht eingesetzt.

Die bekannteste Anwendung ist Scratch, das vom MIT, inspiriert von LOGO (Papert Seymour, 1999), mit dem Ziel eine farbigere, aussagekräftigere und sozialere Programmierumgebung zu schaffen (Resnik M. et al., 2009), entwickelt wurde. Weitere Beispiele sind: Snap, App Inventor, Babuino, Blockly Games, Gamefroot, GameSalad, Drupal, Hopscotch, Modkit, Tynker, Pencil Code, Alice, Open Robert, Stencyl und Pocket Code.

Gründe blockbasierte Programmierumgebungen im Unterricht einzusetzen

Das Programmieren lernen birgt einige Schwierigkeiten. Lernende müssen verstehen wozu Programmierung eigentlich gut ist, welche Vorteile sie haben, wenn sie diese Fähigkeit beherrschen, wie Computersysteme arbeiten, Syntax und Semantik beherrschen, Problemlösungspläne entwerfen, Fähigkeiten entwickeln, um Fehler zu finden und zu beheben. (Bouley B. du, 1989). Wobei Kelleher und Pausch den Syntaxproblemen die größte Bedeutung beimessen. (Kelleher Caitlin & Pausch Randy, 2005)

Blockbasierte Programmierumgebungen weisen einige Vorteile auf die helfen Anfangsschwierigkeiten zu überwinden.

Arbeiten mit Ereignissen

Die Ausführung eines Befehls oder einer Prozedur ist von Ereignissen abhängig. Dies ist für Einsteiger in die Programmierung leichter zu verstehen als Klassen, Objekte, Listeners etc. um ein Ereignis auszudrücken.

Entdecken statt Erinnern

Die Auswahl eines Befehls aus einer vorgegebenen Palette an „Puzzlesteinen“ ist wesentlich einfacher, als das Erlernen und spätere Erinnern hunderter Befehle einer textuellen Programmiersprache. Weiters sind die Blöcke in Kategorien zusammengefasst und mit unterschiedlichen Farben versehen. Statt des Erinnerns an ein zuvor erlerntes Vokabular kann man somit in, nach Funktionalität organisierten Listen, nach einem Block Ausschau halten. (vgl. Bau David, et al., 2017)

Vermeiden von Syntaxfehlern und bessere Lesbarkeit des Codes

Die Form der Blöcke legt nahe, wie sie zusammengesetzt werden sollen, (Resnik M. et al., 2009) es ist meist nicht möglich Teile zu verbinden, bei denen ein Zusammenbau keine sinnvolle Programmstruktur ergeben würde, da diese dann keine Verbindungsstellen aufweisen. Oft werden Blöcke auch schon mit Parameterwerten bereitgestellt oder diese sind mittels Drop-Down auswählbar. (vgl. Bau David et al., 2017) Auch die Formulierungen auf den Blöcken sind meist so gewählt, dass sie einen niederschweligen Einstieg in die Programmierung gewährleisten.

Durch die spezielle Form- und Farbgebung wird der geschriebene Code auch leichter lesbar. Es fällt leicht einzelne Programmelemente zu erkennen und durch die teilweise umschließende Form der Blöcke sieht man auf einen Blick wo eine Schleife oder Prozedur beginnt und wo sie endet.

Entlastung des Kurzzeitgedächtnisses

Befehle, welche bei textbasierten Programmiersprachen aus einer oft beträchtlichen Anzahl von Zeichen und Vokabeln bestehen, werden mittels eines einzigen Blockes dargestellt. Das hilft die kognitive Belastung zu reduzieren, denn wie bereits der amerikanische Psychologe George A. Miller 1956 festgestellt hatte, liegt das unmittelbare Erinnerungsvermögen des Mensch bei nur 7 plus minus 2 Informationseinheiten. (Miller G. A., 1956). (siehe Bezug zu den Lerntheorien - Cognitive load Theorie)

Lernen durch Experimentieren

Da der Code meist sofort ausgeführt wird, sobald das entsprechende Ereignis eintritt, können die Anwender und Anwenderinnen einfach etwas „zusammenklicken“ und unmittelbar kontrollieren, was passiert. Dabei müssen sie Strategien entwickeln um Probleme zu lösen oder dieses zu umgehen. (Brennan K. and Resnick M., 2012). Es können auch Blockteile im Skriptbereich, für eine eventuell spätere Verwendung, ungenutzt liegen gelassen werden. Dies soll zum Experimentieren einladen. (Resnik M. et al., 2009)

Die Erstellung eines Projekts ist kein rein linearer Ablauf, bei dem Schülerinnen und Schülern erst einen Plan entwerfen, ihr Design gestalten und anschließend den Code implementieren. Es ist eher ein adaptiver Prozess, bei dem die Pläne oftmals erst in kleinen Schritten entwickelt werden können. Sie lernen durch das Zusammenstellen von kleineren Teilen zu einem Ganzen das Abstrahieren und Modularisieren, was für jeden Problemlöseprozess wichtig ist. (vgl. Brennan K. and Resnick M., 2012)

Soziales Lernen und Lernen durch Teilen

Viele der o.a. Programme bieten die Möglichkeit Projekte zu veröffentlichen oder zu teilen. Dadurch stehen den Anwenderinnen und Anwendern viele Beispiele zur Verfügung aus denen sie lernen und sich inspirieren lassen können. Dies ermöglicht ihnen komplexere Projekte umzusetzen (vgl. Brennan K. and Resnick M., 2012) und fördert das Lernen durch Experimente sowie soziales Lernen durch eine aktive Community. (vgl. Dumke, Rainer; 2013, p. 259) Zusätzlich wird dabei das Lesen von Code nebenläufig erlernt.

In der Schule kann dieses Feature auch genutzt werden um Quellenangaben, Lizenzen und Open Source zu thematisieren.

Natürlich haben blockbasierte Programmierumgebungen auch Nachteile. Bei umfangreicheren Programmen wird, aufgrund der Größe der Blöcke, der Code oft unübersichtlich und das Schreiben von textuellem Code ist für einen geübten Programmierer schneller, als das Suchen, Auswählen und Zusammensetzen von Blöcken. Diese Punkte spielen jedoch speziell im Anfangsunterricht kaum eine Rolle und werden von einigen Anwendungen bereits in der Form verringert, dass man Blöcke zusammenfallen lassen und verkleinern oder diese mittels Texteingabe finden kann. Beim App

Inventor zum Beispiel erhält man eine Liste der möglichen Befehle, sobald man im Viewer zu schreiben beginnt. Man braucht also zum Beispiel nur noch auf die Taste „W“ zu tippen und erhält eine Liste aller zu diesem Zeitpunkt möglichen „when-Befehle“. Aus dieser Liste wählt man den passenden Befehl aus und muss auf diese Weise nicht alle Blöcke der entsprechenden Kategorie durchsuchen.

Schülergerechte Projekte und Personalisierungsmöglichkeiten

Die meisten blockbasierten Anwendungen bieten die Möglichkeit ein breites Spektrum an Projekten umzusetzen. Es können Spiele, Animationen, Rätsel, Simulationen, Berechnungsmöglichkeiten für mathematische Aufgaben etc. erstellt werden. Somit können unterschiedliche Interessen bedient werden. Auch die vielfältigen Möglichkeiten ein Projekt zu personalisieren, indem man eigene Bilder, Töne und selbst erstellte Grafiken importiert (Peppler K. & Kafai Y., 2007) motiviert und stärkt die Identifikation mit dem Projekt.

Dies in Kombination bietet Lernenden an persönlich bedeutungsvollen Projekten zu arbeiten und dafür sind diese oft bereit wichtige mathematische und rechnerische Konzepte zu erlernen. (Resnik M. et al., 2009)

Lernende können in visuell aufbereiteten und interaktiven Umgebungen arbeiten und können Programme erstellen die sie bereits kennen oder selbst verwenden, was sie vermutlich lieber machen, als Texte in einen Editor einzugeben, mit Fehlermeldungen beim kompilieren konfrontiert zu werden, um am Ende ein Programm zu haben das Zeichen oder Zeichenketten manipuliert. (Landerer Claudio, 2006)

Unterschiedliche Repräsentationsformen

Es wird empfohlen, im Unterricht Denkoperationen wann immer das möglich ist, auf mehreren Ebenen zu ermöglichen. (vgl. Hartman Werner, et.al , 2006, p. 116). Mögliche Repräsentationsformen sind nach Brunner:

- **enaktiv** – durch Handeln
- **ikonisch** – durch Bilder
- **symbolisch** – durch Symbole wie zum Beispiel Sprache (vgl. Bruner J, et al, 1988, p. 27 ff.)

Blockbasierte Programmierungsumgebungen verbinden Elemente ikonischer, symbolischer und enaktiver Repräsentationsformen da die Programme und Blöcke ikonisch dargestellt sind, die Puzzleteile von Anweisungen, Schleifen, Operatoren etc. unterschiedlich aufgebaut und mittels verschiedener Farben visualisiert sind und die textbasierte Programmierung in den Bausteinen enthalten ist. Schlüsselwörter der Programmierung werden so mit symbolischer Darstellung kombiniert. Indem man Programmier-Anweisungsblöcke zusammensetzt, so wie man LEGO-Steine oder Puzzleteile zusammensetzen kann (Resnik M. et al., 2009) kommt auch die enaktive Repräsentation zum Tragen.

Vermittlung wichtiger Konzepte der Programmierung

Bei der Arbeit mit einer blockbasierten Programmiersprache wird der Anwender mit vielen Konzepten, welche auch in vielen anderen Programmiersprachen üblich sind, konfrontiert.

Am Beispiel von Scratch haben Brennan und Resnick sieben Konzepte identifiziert die in vielen Scratchprojekten wichtig und auf andere Programmierkontexte übertragbar sind: (vgl. Brennan K. and Resnick M., 2012)

1. **Sequenzen:** Sind die einfachste Form einer Programmstruktur. Eine lineare Abfolge von Schritten oder Anweisungen, welche vom Computer abgearbeitet werden soll, um eine Aufgabe auszuführen.
2. **Schleifen:** Bieten die Möglichkeit, eine Sequenz mittels einer Bedingung mehrmals auszuführen.
3. **Ereignisse:** Ein bestimmtes Ereignis löst etwas aus.
4. **Parallelität:** Es ist möglich mehrere Sequenzen gleichzeitig ablaufen zu lassen.
5. **Konditionen:** Stellen eine Möglichkeit dar, auf Grundlage bestimmter Bedingungen Entscheidungen zu treffen.
6. **Operatoren:** Liefern eine Unterstützung um mathematische, logische und String Ausdrücke zu manipulieren.
7. **Daten:** Es können Werte erfasst, gespeichert, aktualisiert und wieder abgerufen werden. Bei diesen Werten kann es sich sowohl um einzelne Zahlen als auch um eine Sammlung von Zeichen oder Zeichenfolgen handeln.

Neben den wichtigen Konzepten der Programmierung werden durch das Arbeiten mit blockbasierten Programmiersprachen auch ein besseres Verständnis für digitale Anwendungen, logisches Denken, Problemlösungskompetenzen und Projektmanagementfähigkeiten vermittelt.

Dies trägt wesentlich bei, das Computational Thinking (siehe Computational Thinking) der Anwender zu trainieren.

Studien, wie jene von Price & Barnes (Price Thomas W. & Barnes Tiffany, 2015) sowie von Weintrop & Wilensky (Weintrop David & Wilensky Uri, 2017) belegen, dass blockbasierte Programmierumgebungen den Einstieg in die Programmierung erleichtern.

Computational Thinking

Unser Alltag wird zunehmend von Technologien bestimmt, welche viele Menschen weder verstehen noch daran glauben, dass sie auf diesen Einfluss nehmen können. (vgl. Bandura A., 2001, p. 17) Daher ist es wichtig, dass Computational Thinking bereits in jungen Jahren vermittelt wird.

Das „Coding“ ist ein wichtiger Teil des Computational Thinkings. Programmierkenntnisse befähigen uns die Funktionsweisen, Möglichkeiten und Auswirkungen der Informationstechnologie besser zu verstehen. (Futschek, Gerald; 2016)

Algorithmen stellen eine Schlüsselqualifikation zur Entwicklung der Fähigkeit die Welt als eine ständige Umwandlung von Inputs in Outputs und als algorithmisch kontrolliert zu interpretieren. (vgl. Denning P. J., 2009, p. 29)

Brennan und Resnick fanden heraus, dass Kinder, welche Computerkonzepte verwenden auch bestimmte Perspektiven, wie zum Beispiel soziale Bindungen und den kreativen Ausdruck positiv verändern. (Brennan K. and Resnick M., 2012)

Elektronische Verarbeitungssysteme und Computer (ob in Form von Smartphones, Smartwatches, Tablets, Navigationssystemen ...) sind feste Bestandteile unseres Alltags geworden und ohne sie wären Fortschritte in der Forschung und Wirtschaft kaum noch möglich, was verdeutlicht welche Bedeutung Computational Thinking auch abseits von Informatik hat. Da wir alle von dieser Entwicklung betroffen sind, ist die Auseinandersetzung mit diesem Thema in der Bildung eine nicht ganz neue aber wesentliche Herausforderung. Die Jugendlichen müssen entsprechend ausgebildet werden, um in einer von Technologie bestimmten Welt bestehen zu können. (vgl. Wing J.M., 2006)

Den Begriff Computational Thinking wurde bereits 1980 von Seymour Papert in seinem Buch *„Mindstorms: Children, computers and powerful ideas“* verwendet. Die zurzeit wohl gängigste Definition dieses Begriffs stammt von Jeannette M. Wing und lautet:

„computational thinking is the thought processes involved in formulating a problem and expressing its solution(s) in such a way that a computer—human or machine—can effectively carry out“ (Jeannette M. Wing, 2014)

Es handelt sich hier um die Denkprozesse bei der Formulierung eines Problems um dieses rechnerisch lösen zu können und beeinflusst die Forschung in nahezu allen Disziplinen, sowohl in den Natur- als auch in den Geisteswissenschaften. (Bundy A., 2007)

Die „International Society for Technology in Education & Computer Science Teachers Association“ (im folgenden kurz ISTE & CSTA) veröffentlichte folgende Definition:

- Probleme so zu formulieren, dass wir einen Computer nutzen können, um sie zu lösen
- Logische Analyse und Organisation von Daten
- Darstellung von Daten durch Vereinfachung wie Modelle und Simulationen
- wiederholbares Lösen von Problemstellungen durch algorithmisches Denken
- Identifizieren, Analysieren und Implementieren von realisierbaren Lösungen mit dem Fokus, die effizienteste und effektivste Kombination von Schritten und Ressourcen zu erreichen
- Generalisierung und Übertragung dieses Problemlösungsprozesses auf eine große Vielfalt von Problemen (vgl. „ISTE: computational-thinking-operational-definition-flyer.pdf,“), (vgl. Futschek, Gerald; 2016)

Ebenfalls wird von der ISTE & CSTA ein Vokabular von neun Computational Thinking Fertigkeiten vorgeschlagen: Datensammlung, Datenanalyse, Datenrepräsentation, Problemzerlegung, Abstraktion, Algorithmen und Prozeduren, Automatisierung, Simulation und Parallelisierung. (vgl. Barr, V., Stephenson, C., 2011, p. 48)

Zusätzlich werden von der ISTE & CSTA eine Reihe von Einstellungen angegeben, welche die o.a. Fertigkeiten verbessern und unterstützen können:

- Umgang mit Komplexität
- Beharrlichkeit bei der Arbeit mit schwierigen Problemen
- Toleranz für Mehrdeutigkeiten
- Die Fähigkeit, mit offenen Problemen umzugehen
- Die Fähigkeit, mit anderen zu kommunizieren und mit ihnen zu arbeiten, um ein gemeinsames Ziel oder eine gemeinsame Lösung zu erreichen („ISTE: computational-thinking-operational-definition-flyer.pdf,“)

Jeannette Wing fordert, dass zu den grundlegenden Kompetenzen Lesen, Schreiben und Rechnen, Computational Thinking jedem Kind vermittelt werden soll. In diesem Zusammenhang spricht sie von drei Grundfähigkeiten:

1. **Problemzerlegung:** komplexe Probleme in beherrschbare Teilprobleme zerlegen
2. **Abstraktion:** Probleme verallgemeinern und überprüfen, ob bereits gefundene Lösungen auf ähnliche Probleme übertragbar sind
3. **Algorithmen verwenden:** Probleme mit einer Abfolge von Schritten lösen (vgl. Wing J.M., 2006)

Das österreichische Bildungssystem hat auf die Herausforderung den Schülern und Schülerinnen Computational Thinking zu vermitteln mit einem Pilotprojekt „Digitale Grundbildung“ für das Schuljahr 2017/18 an 169 Neuen Mittelschulen und AHS-Unterstufen reagiert, welches per Verordnung des Lehrplans 2018/19 flächendeckend umgesetzt werden soll. (vgl. Bundesministerium für Bildung, Wissenschaft und Forschung)

Im Rahmen dieses Projekts sollen die Schülerinnen und Schüler unter anderem bereits lernen mit Algorithmen zu arbeiten, einfache Programme zu erstellen und Programmiersprachen kreativ zu nutzen. (vgl. „Verbindliche Übung „Digitale Grundbildung“ in Sekundarstufe 1. Inhalte für Pilotierung im Schuljahr 2017/18,“)

PRAKTISCHER TEIL

App Inventor

Entstehungsgeschichte

App Inventor ist eine blockbasierte Programmierumgebung, welche es ermöglicht mobile Apps für Androidgeräte zu entwickeln.

Ursprünglich wurde der App Inventor von Google unter der Leitung des MIT Professors Hal Abelson und dem Google-Programmierer Mark Friedman entwickelt.

Im Jahre 2009 fand ein Google-Pilotprogramm an 20 Universitäten mit den Zielen

- die Entwicklung mobiler Anwendungen für jedermann zugänglich zu machen
- Lernerfahrungen für EDV-Einsteiger mithilfe der offenen Plattform Android zu erleichtern
- das System als Reaktion auf die Erfahrungen und dem Feedback weiter zu gestalten

und der Aufforderung an die Universitäten und Studierenden sich über Materialien für Lehre und Forschung auszutauschen statt. (Abelson Hal, 2009)

Im Jahr 2010 wurde der App Inventor dann der Öffentlichkeit präsentiert. Ein Jahr später wählte Google das „Center for Mobile Learning“ aus um den öffentlichen Server des App Inventors zu hosten und den Source-Code bereitzustellen. (Clark Andrew, 2013)

In weiterer Folge wurde das Google Labs geschlossen und der App Inventor wechselte zum MIT-Zentrum für mobiles Lernen, weiterhin unter der Leitung von Hal Abelson unter der Mitarbeit der MIT Professoren Eric Klopfer und Michael Resnik.

2012 wurde dann die erste MIT-Version veröffentlicht. (Clark Andrew, 2013)

Schließlich kam es am 6. Dezember 2013 zur Veröffentlichung des MIT-App Inventor 2 und die ursprüngliche Version wurde in „App Inventor Classic“ umbenannt, welche jedoch im Juli 2015 offline ging. („App Inventor Classic | Explore MIT App Inventor“)

Musste man beim App Inventor Classics noch eine Java-Datei installieren, so läuft der App Inventor 2 nun vollständig im Browser. Auch beinhaltet der App Inventor 2 zahlreiche Verbesserungen im Bereich der Usability sowie viele ästhetische Veränderungen.

Das MIT App Inventor-Projekt hat zum Ziel die Softwareentwicklung zu demokratisieren, indem sie alle Menschen, aber vor allem Kinder, unterstützt vom Technologiekonsumenten zum Technologieerzeuger zu werden.

Seit seiner Einführung haben sich über sechs Millionen Benutzer aus 195 Ländern für den Zugriff auf App Inventor registriert und es wurden rund 22 Millionen Apps erstellt. ([“About Us | Explore MIT App Inventor,”](#))

App Inventor ist inspiriert von Scratch. Beide Anwendungen verwenden die gleiche Code-Bibliothek „Blockly“ um ihre Blöcke im Browser zu erstellen.

Besondere Features des App Inventors

Neben den Vorteilen blockbasierter Programmierumgebungen bietet der App Inventor zahlreiche weitere Möglichkeiten.

Mithilfe dieser Anwendung ist es den Lernenden möglich, personalisierte Apps für ihr eigenes Smartphone oder Tablett zu erstellen. Sie können ihrer Kreativität freien Lauf lassen, Daten teilen, Funktionen ihres Handys nutzen und etwas schaffen das Spaß macht. Ihr Werk können sie dann Freunden, Bekannten und der Familie jederzeit präsentieren, da sie es immer mit sich führen.

Da App Inventor beinhaltet komplexe Technologien zur Be- und Verarbeitung von SMS-Nachrichten, E-Mails und Daten der GPS-Ortungssensoren und der NFC-Funktion des Telefons, zum Scannen von Barcodes sowie zur Kommunikation mit Web-API's und Twitter.

Ebenfalls wird der Zugriff auf sämtliche Telefon- und Kamerafunktionen des Smartphones inklusive Telefonbuch und Bilddateien unterstützt.

Es kann auch auf den Beschleunigungssensor des Handys zugegriffen werden, womit man weitere Möglichkeiten der Steuerung erhält und Schrittzähler implementieren kann.

Hier gilt es anzumerken, dass die meisten dieser Funktionen nicht durch den Emulator unterstützt werden und aus diesem Grund nur am eigenen Androidgerät getestet werden können.

Zur Speicherung und Verarbeitung von Daten stehen „File“, „TinyDB“, „TinyWebDB“ und CloudDB zur Verfügung. Diese ermöglichen persistentes Speichern mit einfachen Schlüssel-Wert-Paaren.

File und TinyDB wird verwendet, um Daten direkt auf dem Gerät zu speichern. Diese Daten sind bei jeder Ausführung der App verfügbar. Auf diese Weise können zum Beispiel Spielstände einfach gespeichert werden.

TinyWebDB bietet die Möglichkeit Daten in einer Webdatenbank zu speichern.

Mittels CloudDB können Daten auf einem mit dem Internet verbundenen Datenbankserver gespeichert werden. Dies ermöglicht den Benutzern einer App, Daten miteinander zu teilen. Standardmäßig werden Daten auf einem Server gespeichert, der von MIT verwaltet wird. Es kann sogar ein eigener Server eingerichtet werden.

Die zur Verfügung stehenden Datenbanken sind vor allem für Multi-User-Anwendungen von großem Nutzen.

„FusionalControl“ ermöglicht die Kommunikation mit Google Fusion tables und somit das Speichern, Freigeben, Abfragen und Visualisieren von Datentabellen.

Es besteht auch die Möglichkeit Apps zur Steuerung von LEGO-Mindstorms zu entwickeln.

Der App Inventor verfügt darüber hinaus über eine Komponente zum Übersetzen von Wörtern und Sätzen in verschiedene Sprachen. Um diese Komponente nutzen zu können braucht man allerdings eine aktive Internetverbindung, da diese eine Verbindung zum Übersetzungsdienst Yandex benötigt.

Ein besonderes Feature ist das „Live-Testing“. Schon während der Entwicklung kann ein Smartphone oder Tablett mit dem App Inventor verbunden werden. So kann jede Änderung in Echtzeit, ohne weitere Schritte durchführen zu müssen, getestet werden. Diese Funktion hilft vor allem beim Gestalten der Benutzeroberfläche, da man sofort sieht, was eine Veränderung bewirkt.

Als besonders positiv hervorzuheben ist auch die Tatsache, dass kein aktives Speichern während der Arbeit erforderlich ist, da der App Inventor jede Änderung automatisch speichert. Auch wird am Computer kein Speicherplatz benötigt. Das komplette Programm und auch die zusätzlich hochgeladenen Medien werden in der Cloud des App Inventors abgelegt.

Das Teilen und Veröffentlichen geht einfach. Das Programm kann entweder als ausführbare *apk.Datei* oder als Sourcecode in Form einer *aia.Datei* auf den eigenen Computer heruntergeladen werden. Eine detaillierte Beschreibung befindet sich im Unterrichtskonzept. (siehe Unterrichtskonzept)

Apps die mit dem App Inventor erstellt werden können

Der App Inventor bietet so viele Möglichkeiten, dass man zumindest im Rahmen des Unterrichts kaum an Grenzen stoßen wird. Über *Gallery* im Hauptmenü des App Inventors kommt man zu den Tutorien und findet auch eine Vielzahl an Beispielen zur Inspiration.

Es ist möglich Spiele, Lernapps, Lokationsanwendungen mittel GPS, technisch hochwertige Anwendung unter Verwendungen der Sensoren des Telefons, Nachrichten- und Chatapps, Anwendungen zur Steuerung der LEGO-Mindstorm-Roboter, Übersetzungsapps, Anwendungen welche mit dem Internet Daten austauschen, Multi-User-Anwendungen und vieles mehr zu entwickeln.

Erscheinungsbild des App Inventors

Der App Inventor besteht im Wesentlichen aus zwei Hauptansichten zwischen denen jederzeit umgeschaltet werden kann, dem Design-Editor, indem das User Interface der Anwendung erstellt wird und dem Block-Editor zum Definieren des Verhaltens der einzelnen Komponenten. Das Hauptmenü ist in beiden Bereichen zugänglich.

Design-Editor

Es stehen viele unterschiedliche Komponenten zum Design der Benutzeroberfläche zur Verfügung. Für formularbasierte Apps gibt es unter anderem Buttons, Labels, Textfelder, Checkboxes, Auswahllisten, Slider und Warndialoge. Leinwand, Bilder und Sprite Komponenten werden für die Erstellung grafischer und animierter Anwendungen verwendet. Diverse Mediakomponenten ermöglichen das Einbinden von Bildern, Videos und Sounddateien.

Die Eigenschaften der gewählten Komponenten wie Größe, Farbe, Beschriftung, Startwert, Intervalle und vieles mehr, können in der Spalte *Properties* angepasst werden.

Im Design-Editor werden auch die nicht-visuellen Komponenten ausgewählt welche die App für den Zugriff auf die auf dem Android-Gerät verfügbare Technologie benötigt.

Die Komponenten werden in der Spalte *Palette* ausgewählt und per Drag-and-Drop auf den virtuellen Bildschirm im *Viewer* gezogen. Dieser besteht aus beliebig vielen *Screens*. Dort können sie mithilfe der Layout-Komponenten, welche ebenfalls in der Spalte *Palette* zu finden sind, nach Belieben

angeordnet werden. Während dieses Vorganges sollte nicht auf das „Live-Testing“ verzichtet werden, da das Erscheinungsbild am Endgerät möglicherweise etwas von der Ansicht im *Viewer* abweicht.

In der Spalte *Components* sieht man eine Liste aller ausgewählten Komponenten des aktuellen *Screens*. An dieser Stelle hat man die Möglichkeit passende Namen zu vergeben oder einzelne Komponenten wieder zu löschen.

Der direkt darunter liegende Abschnitt *Media* liefert eine Übersicht aller verwendeten Musik- Video- und Bilddateien und bietet die Möglichkeit weitere solche Dateien hochzuladen.

Block-Editor

Der Blockeditor hat zwei Bereiche. Der Erste, *Blocks*, enthält die zur Programmierung des Verhaltens der App notwendigen Blöcke. In *Built-in* sind die grundlegenden Blöcke der App Inventor Sprache enthalten. Hierbei handelt es sich um Anweisungen und Kontrollstrukturen wie Schleifen, Verzweigungen, Variablen, mathematische und logische Operatoren, Blöcke zur Textverarbeitung, Listen, Farben und vorgefertigte Prozeduren. Sobald im Design-Editor eine Komponente hinzugefügt wurde, wird im Abschnitt des entsprechenden *Screens* ein Satz von Komponentenblöcken hinzugefügt, welche dann Aktionen für diese Komponente ausführen können. Im Bereich *Any component* sind Blöcke zur Interaktion mit beliebigen Komponenten der Anwendung enthalten.

Unter dem Bereich *Blocks* findet man bei *Media* erneut die verwendeten Audio-, Video- und Bilddateien und hat auch hier die Gelegenheit zum upload weiterer Dateien.

Alle Blöcke bieten visuelle Hinweise, um die Entwicklungsarbeit zu erleichtern und können aufgrund ihrer Form leicht zu syntaktisch korrektem Code zusammengefügt werden.

Dies geschieht im zweiten Bereich. Hierher werden die Blöcke mittels Drag-and-Drop aus der Spalte *Blocks* gezogen und zu einem Programm zusammengesetzt.

Eine detaillierte Beschreibung der technischen Voraussetzungen, des Design- und Blockeditors befindet sich im Unterrichtskonzept.

Unterrichtskonzept zu Einführung in die Programmierung

Im Rahmen dieser Diplomarbeit wurde ein Unterrichtskonzept zur „Einführung in die Programmierung mit dem App Inventor“ erstellt. Mit diesem Konzept wird versucht sowohl die für die vom Lehrplan vorgesehenen informatischen Kompetenzen als auch die für die Informatik wesentlichen sozialen Kompetenzen zu vermitteln.

Als Erstes wird der Bezug zum Lehrplan hergestellt. Im weiteren Verlauf werden die Methode und die Ergebnisse der Evaluation des Unterrichtskonzeptes, durch erfahrene Lehrpersonen, präsentiert, sowie die daraus resultierenden Änderungen am Konzept beschrieben.

Bezug des Konzeptes zum Lehrplan

Das Unterrichtskonzept dieser Arbeit ist darauf ausgerichtet direkt und indirekt eine Vielzahl der im Lehrplan erwähnten Kompetenzen vermitteln zu können.

Unter Kompetenzen versteht mal laut Franz E. Weinert

„[...] die bei Individuen verfügbaren oder durch sie erlernbaren kognitiven Fähigkeiten und Fertigkeiten, um bestimmte Probleme zu lösen, sowie die damit verbundenen motivationalen und sozialen Bereitschaften und Fähigkeiten, um die Problemlösung in variablen Situationen erfolgreich und verantwortungsvoll nutzen zu können“. (Weinert; Franz E. (Hrsg), 2001, p. 27 f.)

Der Bezug zum Lehrplan, für die sozialen Kompetenzen, welche mit diesem Konzept durch das verstärkte Augenmerk auf das Kommunikationsverhalten der Schülerinnen und Schüler, sowie die Auseinandersetzung mit der personenzentrierten Kommunikation und dem aktiven Zuhören, vermittelt werden, wurde schon im theoretischen Teil dieser Arbeit behandelt.

Es ist ein weiterer Bestandteil des Konzepts, dass sich die Schülerinnen und Schüler mit den Inhalten zu einem großen Teil, unter Nutzung digitaler Medien und von der Lehrperson, in digitaler Form zur Verfügung gestellten Unterlagen, Wissen und Fähigkeiten selbstständig aneignen. Dies soll dazu beitragen, dass neben den im Lehrplan für Informatik in der 5. Klasse AHS Oberstufe genannten fachlichen Kompetenzen der angewandten und praktischen Informatik, wie:

- *„Informationsquellen erschließen, Inhalte systematisieren, strukturieren, bewerten, verarbeiten und unterschiedliche Informationsdarstellungen verwenden können“*
- *„Begriffe und Konzepte der Informatik verstehen und Methoden und Arbeitsweisen anwenden können“*
- *„Algorithmen erklären, entwerfen, darstellen und in einer Programmiersprache implementieren können“*

- „Grundprinzipien von Algorithmen, Datenstrukturen und Programmen erklären können“
(“RIS - Lehrpläne – allgemeinbildende höhere Schulen - Bundesrecht konsolidiert,
Fassung vom 18.03.2018,”)

auch die überfachlichen Kompetenzen erarbeitet werden.

So wird durch die freie Wahl betreffend der Gestaltung der Umgestaltungsaufgaben in den **Einführungsbeispielen**, sowie beim **Abschlussprojekt** die Forderung des o.a. Lehrplans, dass Schülerinnen und Schüler ihr kognitives, emotionales und kreatives Potenzial nützen sollen, erfüllt. (vgl. ebd.)

In den „*didaktischen Grundsätzen*“ des Lehrplans für Informatik ist weiters erwähnt, dass das Unterrichtsmanagement „[...] *sich an für Schülerinnen und Schüler transparenten Lehrzielen zu orientieren*“ hat und „[...] *beispielgebend für die eigene Lern- und Arbeitsorganisation auch außerhalb des Informatikunterrichts sein*“ soll. Diesem Punkt wird im einführenden **Gruppengespräch** und in den regelmäßigen **Runden** zum Informationsaustausch Rechnung getragen. Es wird sowohl mittels der **Einführungsbeispiele** und der **Projektarbeit**, als auch in den diversen Gesprächen den Schülerinnen und Schülern „[...] *Gelegenheit [...]*“ gegeben, „[...] *Neues zu erforschen und bereits Gelerntes in verschiedenen kommunikativen und inhaltlichen Kontexten anzuwenden*“, wie dies der Lehrplan vorsieht. Auch wird „*die Bedeutung gemeinschaftlichen Problemlösens bei der Bearbeitung von Projekten [...]*“ thematisiert und bei der Teamarbeit umgesetzt. (ebd.)

Die im Konzept geplanten Gruppengespräche, Präsentationen, das Online-Forum und die Möglichkeit Ergebnisse der Allgemeinheit zur Verfügung zu stellen tragen wie im Lehrplan gewünscht „*zur Motivation und zur Sicherung des Unterrichtsertrages [...]*“ bei, da sie unterschiedliche Möglichkeiten haben „[...] *ihr Wissen zu präsentieren sich der Kritik anderer zu stellen, und ihre Arbeit zu argumentieren.*“ (ebd.)

Den „*Leitvorstellungen*“ des Allgemeinen Lehrplans für die AHS kann man entnehmen, dass die Schülerinnen und Schüler „[...] *zur Übernahme sozialer Verantwortung*“ und „[...] *Urteilsbildung [...]*“ zu befähigen sind und es dazu erforderlich ist, ein hohes Maß an „[...] *Selbstsicherheit sowie selbstbestimmtes und selbst organisiertes Lernen zu fördern*“, und zu „[...] *selbstständiges Handeln als auch zur Teilhabe am sozialen Geschehen anzuhalten*“. Unter dem Punkt „*Aufgabenbereiche der Schule - Wissensaneignung, Kompetenzerwerb*“ ist festgehalten, dass „[...] *die Schülerinnen und Schüler im Sinne eines lebensbegleitenden Lernens zur selbstständigen, aktiven Aneignung, aber auch zu einer kritisch-prüfenden Auseinandersetzung mit dem verfügbaren Wissen befähigt und ermutigt werden sollen.*“ (ebd.). Diese Vorgaben werden mit dem Unterrichtskonzept dahingehend umgesetzt, dass die Schülerinnen und Schüler frei aus dem Informationsangebot wählen können, ihr Arbeitstempo weitgehend selbst festlegen und im Team aber auch als Gruppe zusammenarbeiten. (ebd.)

Im „*allgemeinen Teil*“ des Lehrplans steht unter „*Bildungsbereich Sprache und Kommunikation*“ aber auch, dass „*ein kritischer Umgang mit und eine konstruktive Nutzung von Medien sind zu fördern.*“ (ebd.). Im Unterrichtskonzept ist daher vorgesehen, dass die Schüler und Schülerinnen Recherchen im Internet durchführen und Informationsquellen, welche sie für hilfreich erachten, auf der gemeinsam genutzten Plattform teilen. (ebd.)

Durch die Verwendung des App Inventors wird auch die im Lehrplan geforderte Nutzung der „*[...] Fremdsprache als Arbeitssprache [...] (Content and Language Integrated Learning)*“ gefördert, da dieser nur in englischer Sprache zur Verfügung steht. Auch stellt das Programmieren von Anwendungen für das eigene Smartphone einen Bezug „*[...] zur Lebens- und Begriffswelt der Jugendlichen [...]*“ her. (ebd.) Es wurde auch bei der Auswahl der **Einführungsbeispiele** auf diesen Aspekt geachtet. (ebd.)

Evaluierung des Konzepts

Als Werkzeug zur Evaluierung des Konzepts wurde eine Expertenbefragung mittels Fragebogen durchgeführt. (siehe Methode).

Um die Forschungsfragen beantworten zu können wurde ein Online-Fragebogen erstellt und dieser mit dem Unterrichtskonzept an acht Experten weitergeleitet.

Die Entscheidung zur Online-Befragung wurde getroffen, um den Befragten möglichst wenig Zusatzaufwand zu bereiten. Ein Klick auf den bereitgestellten Link genügt, um den Fragebogen zu öffnen, nach dessen Bearbeitung sind keine weiteren Schritte erforderlich und die Bearbeitung kann zu jedem beliebigen Zeitpunkt durchgeführt werden.

Neben der Beantwortung der Forschungsfragen ist auch die persönlichen Meinungen der Befragten zum Thema „Relevanz des Codings“ von Interesse. Ein wesentliches Anliegen besteht aber darin, Verbesserungsvorschläge und Anregungen für das Konzept zu erhalten.

Auswahl der Experten

Da ich selbst leider noch wenige Kontakte zu Lehrpersonen knüpfen konnte, erhielt ich von Frau ao. Univ.-Prof. Dipl.-Ing. Dr. Renate Motschnig sechs Kontakte zur Verfügung gestellt. Dabei handelt es sich im Lehrerinnen und Lehrer welche allesamt bereits über Unterrichtserfahrung im Fach Informatik verfügen. Nach einer Kontaktaufnahme meinerseits, mit diesen Experten, erklärten diese sich bereit das Konzept zu überprüfen und den bereitgestellten Fragebogen zu bearbeiten.

Weiteren zwei Lehrern, welche ich während meiner fachpraktischen Ausbildung kennenlernen durfte, wurde das Konzept mit der Bitte um Überprüfung und Beantwortung des Fragebogens übermittelt. Bei diesen Personen handelt es sich ebenfalls um Informatiklehrer mit mehrjähriger Berufserfahrung.

Die ausgewählten Experten unterrichten an unterschiedlichen allgemein bildenden höheren Schulen in Wien und Niederösterreich.

Genauere Angaben zu den Experten sind in der Auswertung der Expertenbefragung zu finden.

Konzeption des Fragebogens

Die Erstellung des Fragebogens richtete sich nach den Regeln zur Formulierung von Fragen (vgl. Bortz, Jürgen & Döring, Nicola, 2007, Kapitel 10) für qualitative Befragungen und den Regeln zur richtigen Frageformulierung (vgl. Lang; Sabine) und dem Arbeitsbuch „Fragebogen“ von Rolf Porst (vgl. Porst; Rolf, 2014).

Die o.a. Literaturrecherche führte zur Bemühung die **Fragen** verständlich und zielgruppenorientiert, einfach und direkt, eindeutig und neutral zu formulieren. Weiters wurde nur ein Sachverhalt pro Frage abgefragt und keine hypothetischen oder suggestiven Fragen gestellt. Ebenso wurden „Warum-Fragen“ vermieden und geschlossene Fragen zur Filterung verwendet.

Bei den **Antwortvorgaben** der geschlossenen Fragen wurde versucht, immer alle Möglichkeiten zu berücksichtigen, die Vorgaben disjunkt, nicht wertend sowie gleichwertig zu halten und auf eine angemessene Anzahl zu achten.

Um eine einheitliche Vorgehensweise beim Ausfüllen des Fragebogens zu ermöglichen wurde zum größten Teil das Schulnotensystem (Ordinalskala 1-5) als Antwortvorgabe gewählt.

Bei der **Fragebogengestaltung** wurde auf eine einfache und strukturierte Form geachtet, um die Übersichtlichkeit zu gewährleisten. Auf eine Begrüßung der Experten und die an diese gerichtete Bitte die Fragen kritisch zu beantworten folgt der Hinweis, dass die Befragung anonym ist und alle Angaben freiwillig sind. Um die Freiwilligkeit während der Befragung zu betonen, wurde bei jedem Punkt die Auswahlmöglichkeit „*keine Angabe*“ zur Verfügung gestellt.

Im weiteren Verlauf sind die Fragen nach Themengebieten zusammengefasst. Dabei wurde darauf geachtet, nicht zwischen den Themen hin und her zu springen. Gegen Ende des Fragebogens erhalten die Experten die Möglichkeit in Form einer offenen Frage noch einmal zu den Themen Stellung zu beziehen. Bei der Länge des Fragebogens wurde darauf geachtet, dass dieser alle relevanten Themen beinhaltet, aber trotzdem nicht zu lange wird. Er endet mit einer Danksagung und Verabschiedung.

Aus den Inhalten ließ sich eine große Anzahl an Fragen ableiten, welche dann auf Relevanz, Wichtigkeit und Redundanz untersucht wurden, um eine Reduktion zu ermöglichen, da der Fragebogen nicht zu lange werden sollte, um den Aufwand der Befragten im Rahmen zu halten und somit deren Akzeptanz für den Fragebogen zu erhöhen.

Es wurde ein erster **Entwurf** erstellt. Um den Fragebogen zu strukturieren, wurden die Fragen nach Themen wie „Aufbau und Struktur“, „Beispielaufgaben“ usw. zusammengefasst.

Dieser Entwurf wurde im Anschluss Studienkolleginnen und Bekannten zum **Pretest** bereitgestellt.

Da es sich bei einigen dieser Personen um Informatiklehrer handelte, also um Personen aus der Zielgruppe, und diese ausführlich Feedback gaben, handelt es sich um einen qualitativen Pretest (vgl. Bortz, Jürgen & Döring, Nicola, 2007, p. 411)

Hieraus ergaben sich noch zahlreiche Änderungen, da einige Formulierungen als zu ausführlich erachtet wurden. Im Zuge der Überarbeitung wurden sämtliche Fragen zu Aussagen umformuliert. Das Ziel war klare und unmissverständliche Formulierungen und Aussagen, welche möglichst kurz sind.

Es ergaben sich 15 Fragen, wovon zwei als Möglichkeit zur Ergänzung der vorangestellten Aussage dienen und vier Angaben zur Person abfragen. Die restlichen Kategorien beinhalten je zwei bis sieben Aussagen, welche bewertet werden sollen, sowie eine Entscheidungsfrage und zwei offene Fragen. Die offenen Fragen beziehen sich auf die persönliche Bewertung der Befragten nach der Relevanz des Codings im Schulunterricht und auf Anregungen, Verbesserungsvorschlägen und Kommentaren zum Konzept. Die Entscheidung hauptsächlich geschlossenen Fragen zu verwenden, begründet auf der Tatsache, dass durch dieses Antwortformat die Auswertungsobjektivität erhöht wird. (vgl. Bortz, Jürgen & Döring, Nicola, 2007, p. 35) und der Aufwand für die Befragten sinkt.

Eine letzte Überarbeitung des Fragebogens fand nach einer Präsentation des selbigen in der Lehrveranstaltung „Diplomentenseminar 2 für Informatikerinnen“ statt. Auf Anregung eines Kollegen wurde bei zwei Fragen die Bewertung durch das Schulnotensystem durch „trifft zu, trifft eher zu, trifft teilweise zu, trifft eher nicht zu, trifft nicht zu“ ersetzt um den negativen Aspekt einer eventuell zu vergebenden „schlechten“ Note zu eliminieren.

Die Endversion des Fragebogens befindet sich im Anhang.

Erhebungsinstrument

Online-Befragung über „soSci Survey“. Bei „soSci Survey“ handelt es sich um eine Onlineplattform, zur Durchführung von Onlinebefragungen. Es ist möglich Fragenkataloge mittels verschiedener Fragetypen zu erstellen, diese zu Fragebögen zusammenzufassen und Links zur Befragung zu veröffentlichen bzw. diese den Teilnehmern zu übermitteln.

Für nicht kommerzielle akademische Studien, für öffentliche Einrichtungen und gemeinnützige Organisationen ist die Nutzung ohne Werbung kostenlos.

Es ist möglich Daten so zu erheben, dass keine Rückschlüsse auf die Teilnehmer möglich sind. Beim Zugriff werden weder IP-Adressen gespeichert noch Cookies verwendet. Die Datenübertragung ist verschlüsselt und die Server von „soSci Survey“ befinden sich in München. Der Umfrageersteller kann jederzeit alle erhobenen Daten aktiv löschen. Geschieht dies nicht, werden die Daten nach drei Monaten Inaktivität automatisch gelöscht. (vgl. „SoSci Survey,“)

Rücklauf

Für die Verwertbarkeit der Ergebnisse einer schriftlichen Befragung ist die Zusammensetzung der Stichprobe relevant. (vgl. Bortz, Jürgen & Döring, Nicola, 2007, p. 259) Daher sind im folgenden kurz der zeitliche Ablauf, die Rücklaufquote und die Personendaten, welche mithilfe des Fragebogens erhoben wurden zusammengefasst.

Die Befragung startete am 15. April 2018. Zu diesem Zeitpunkt wurde den potenziellen Teilnehmern der Link zum Fragebogen sowie das Unterrichtskonzept übermittelt.

Einige Tage vor Ablauf der Umfrage, wurde an die Experten ein Schreiben geschickt, indem ich mich für die bereits bearbeiteten Fragebögen bedankte beziehungsweise auf das baldige Ende der Befragung hinwies, um auch all diejenigen, welche bis zu diesem Zeitpunkt noch nicht zur Bearbeitung kamen, die Möglichkeit zu geben diese noch durchzuführen.

Nach Beendigung der Umfrage hatten sechs der o.a. acht Experten an der Umfrage teilgenommen. Aufgrund der Anonymität der Befragung konnte jedoch nicht festgestellt werden, wer genau den Fragebogen beantwortet hat.

Von den Personen wurden lediglich folgende Angaben getätigt:

- Sämtliche Teilnehmer unterrichten das Fach Informatik an einer AHS, zwei davon zusätzlich an einem BORG mit Informatikzweig.
- Nur ein Experte hat weniger als 5 Jahre, die übrigen 5 – 10 Jahre Berufserfahrung.
- Als Zweitfächer wurden Geografie, Psychologie/Philosophie, Mathematik und Geschichte angegeben, zweimal wurde keine Angabe gewählt.

Auswertung der Expertenbefragung

Zur Auswertung der geschlossenen Fragen wurden die Ergebnisse der bearbeiteten Fragebögen tabellarisch erfasst, um eine bessere Übersicht zu gewährleisten. Bei den Bewertungen nach dem Schulnotensystem wurde zusätzlich der Durchschnittswert berechnet.

Die Auswertung der personenbezogenen Fragen wurde bereits im Unterpunkt „Rücklauf“ thematisiert.

Auswertung der geschlossenen Fragen

Aufbau und Struktur des Konzepts

	Anzahl der vergebenen Noten					k. A.	Durchschnitt
	Note 1	Note 2	Note 3	Note 4	Note 5		
Aufbau und Struktur des Konzepts							
Das Konzept ist übersichtlich gestaltet	4	2	-	-	-	-	1,33
Das Konzept ist gut strukturiert	3	3	-	-	-	-	1,50
Das verwendete Bildmaterial ist hilfreich	3	3	-	-	-	-	1,50
Die enthaltenen Links sind hilfreich	4	1	1	-	-	-	1,50
Das Konzept enthält alle benötigten Informationen	5	-	1	-	-	-	1,33

Abbildung 1 - Auswertung Aufbau und Struktur

Der Aufbau und die Struktur des Konzeptes wurden von den Experten als gut empfunden. Auch die bereitgestellten Materialien wurden als hilfreich eingeschätzt.

Beispielaufgaben

	Anzahl der vergebenen Noten					k.A.	Durchschnitt
	Note 1	Note 2	Note 3	Note 4	Note 5		
Bewertung der Beispielaufgaben							
Erklärungen sind gut verständlich	2	3	-	-	-	-	1,33
einzelnen Schritte sind gut nachvollziehbar	1	3	2	-	-	-	1,67
Beispielvideos sind eine gute Ergänzung	5	1	-	-	-	-	1,17
Arbeitsblätter sind hilfreich	3	3	-	-	-	-	1,50
Aufgaben sind verständlich	6	-	-	-	-	-	1,00
Aufgaben sind SchülerInnen gerecht	4	1	1	-	-	-	1,50
Unterlagen ermöglichen ein selbstständiges Arbeiten	2	2	1	1	-	-	2,17

Abbildung 2 - Auswertung Beispielaufgaben

Die Aufgaben wurden als sehr gut verständlich, sowie von der Mehrheit der Experten als schülerinnen- und schülergerecht empfunden. Ebenso wurden die Beispielvideos als eine sehr gute Ergänzung angesehen. Die Verständlichkeit der Erklärungen wurde als gut eingestuft. Auch wurden die Arbeitsblätter als hilfreich angesehen. Geteilt sind die Meinungen der Experten bezüglich der

Frage, ob diese Unterlagen ein selbstständiges Arbeiten der Schülerinnen und Schüler ermöglichen, wobei auch hier die Mehrheit der Meinung ist, dass es gut funktionieren kann. Zwei der Befragten waren nur teilweise der Meinung, dass die einzelnen Schritte in den Beispielaufgaben gut nachvollziehbar sind, der Rest bewertete diesen Punkt als gut.

Einführung in die Programmierung

	Anzahl der vergebenen Noten					k.A.	Durchschnitt
	Note 1	Note 2	Note 3	Note 4	Note 5		
Einführung in die Programmierung in der Schule							
Der App Inventor ist dafür geeignet	1	3	2	-	-	-	2,17
Das Konzept erfüllt die Forderungen der Lehrplans	3	3	-	-	-	-	1,50
Der soziale Mehrwert wird erkannt	-	3	2	1	-	-	2,67

Abbildung 3 - Auswertung App Inventor zur Einführung in die Programmierung

Die Eignung des App Inventors für den Einsatz im Unterricht wurde als durchschnittlich gut bewertet. Lediglich zwei der sechs Experten halten den App Inventor nur bedingt für geeignet, um die Einführung in die Programmierung zu unterrichten. Die Hypothese, dass vor allem jene Experten, welche den App Inventor bereits zur Einführung in die Programmierung eingesetzt haben, diesen als dafür sehr gut geeignet halten, hat sich nicht bestätigt.

Die Befragten bestätigten, dass die vom Lehrplan vorgesehenen Kompetenzen mithilfe dieses Konzepts vermittelt werden können.

Lediglich der soziale Mehrwert, welcher versucht wird, mit diesem Unterrichtskonzept zu vermitteln, wurde nur teilweise erkannt.

Erfahrungen mit dem App Inventor

	Anzahl der vergebenen Noten					k.A.
	trifft zu	trifft eher zu	trifft teilweise zu	trifft eher nicht zu	trifft nicht zu	
Erfahrungen mit dem App Inventor						
kannte den App Inventor schon vorher	3	2	1	-	-	-
habe Erfahrungen mit dem App Inventor im Unterricht	1	1	-	1	3	-

Abbildung 4 - Auswertung Erfahrungen mit dem App Inventor

Der App Inventor war den Experten schon vor Erhalt dieses Konzeptes bekannt. Die Hälfte der Befragten kannte ihn sogar bereits sehr gut. Ein Experte arbeitet bereits regelmäßig mit dem App Inventor im Unterricht und ein weiterer hat auch schon einige Erfahrungen mit dieser Anwendung gesammelt.

Vermittlung sozialer Kompetenzen

	Anzahl der vergebenen Noten					k.A.	Durchschnitt
	Note 1	Note 2	Note 3	Note 4	Note 5		
Vermittlung sozialer Fähigkeiten							
Gruppengespräche sind eine sinnvolle Ergänzung	3	2	-	1	-	-	2,17
Präsentationen sind ein wichtiger Bestandteil	3	3	-	-	-	-	1,50
Reaktionsblätter stellen eine gute Ergänzung dar	2	4	-	-	-	-	1,67
Kommunikation und aktives Zuhören	1	3	1	1	-	-	2,33
Vermittlung von Soft Skills	6	-	-	-	-	-	1,00

Abbildung 5 - Auswertung Vermittlung sozialer Fähigkeiten

Alle Experten stimmen überein, dass die Vermittlung von Soft-Skills im Informatikunterricht wichtig ist. Die im Konzept vorgesehenen Ergänzungen wie Gruppengespräche, Präsentationen und Reaktionsblätter wurden bis auf eine Ausnahme als gut empfunden. Lediglich ein Experte sah in den Gruppengesprächen wenig Sinnvolles. Am wenigsten Zuspruch erhielt die Idee, Kommunikation und aktives Zuhören explizit zu thematisieren.

Verwendbarkeit

	Anzahl der vergebenen Noten					k.A.	Durchschnitt
	Note 1	Note 2	Note 3	Note 4	Note 5		
Verwendbarkeit							
Konzept ist eine gute Unterstützung	4	2	-	-	-	-	1,33
werde das Konzept selbst einsetzen	3	-	1	1	-	1	1,67
werde das Konzept weiterempfehlen	5	-	1	-	-	-	1,33

Abbildung 6 - Auswertung Verwendbarkeit

Das Konzept wurde als eine gute Unterstützung für den Informatikunterricht angesehen. Drei der Experten werden vermutlich dieses Konzept selbst im Unterricht verwenden und fünf werden es Kolleginnen und Kollegen weiterempfehlen.

Ergänzungen

	Anzahl der vergebenen Noten		
	ja	nein	k.A.
Das Konzept gehört noch ergänzt	1	3	2

Abbildung 7 - Auswertung Ergänzungen

Ein Experte findet, dass das Konzept noch ergänzt gehört, weiter drei sind der Meinung, dass keine weiteren Ergänzungen notwendig sind. Die Wünsche der Experten zu möglichen Ergänzungen werden bei der Auswertung der offenen Fragen thematisiert.

Rolle der Lehrperson

	Anzahl der vergebenen Noten					k.A.
	trifft zu	trifft eher zu	trifft teilweise zu	trifft eher nicht zu	trifft nicht zu	
Rolle als Lehrperson						
Ich sehe mich eher als Instruktor	-	2	3	1	-	-
Ich sehe mich eher als Lernbegleiter	1	3	2	-	-	-
kann mit gut vorstellen in die Rolle des Begleiters zu schlüpfen	3	-	-	-	-	3

Abbildung 8 - Auswertung Rolle der Lehrperson

Nur eine der befragten Lehrpersonen sieht sich schon jetzt als Lernbegleiter. Der Rest sieht sich selbst sowohl als Lernbegleiter als auch als Instruktor. Jedoch können sich drei der Befragten vorstellen in Zukunft diese Funktion zu übernehmen, die übrigen nahmen dazu nicht Stellung, wobei anzumerken ist, dass ein Experte der sich dazu nicht äußert, sich bereits als Lernbegleiter wahrnimmt.

Auswertung der offenen Fragen

Vorgehen

Bei den offenen Fragen ergab sich insofern ein einheitliches Bild, als dass die Experten lediglich eine dieser Möglichkeiten nutzten um Gedanken und Anregungen zu formulieren.

Aus diesem Grund wurden die Antworten aus den folgende offenen Fragen:

- Das fehlt mir an diesem Konzept
- Warum finden Sie es wichtig bzw. nicht so wichtig Coding in der Schule zu unterrichten
- Was ich zu meiner Rolle als Lehrperson noch sagen möchte
- Verbesserungsvorschläge, Hinweise und Anregungen

in einem Textdokument zusammengefasst und nach den wesentlichen Grundsätzen der qualitativen Inhaltsanalyse unterzogen. Diese formuliert Regeln nach denen die Texte analysiert werden, welche sich an die Fragestellung anpassen. Aus Modellen wie Zusammenfassung, induktiver Kategorienbildung, Explikation und Strukturierung wird das passende bzw. eine Kombination daraus gewählt. Die Kategorienbildung erfolgt zirkulär. (vgl. Mayring; Philipp, 2010)

Im konkreten Fall wurden die Aussagen, wie oben beschrieben zusammengefasst und in einem weiteren Schritt in Kategorien aufgeteilt. Dazu wurden konkrete Textstellen ausgewählt, welche unter eine Kategorie fallen. Anschließend wurden die Texte in den einzelnen Kategorien erneut zusammengefasst. Dabei gelang eine erste Reduktion der Texte. In weiter Folge wurde das Kategoriensystem überarbeitet und der Vorgang mehrmals wiederholt.

Erst dann wurden die einzelnen einer quantitativen und qualitativen Analyse unterzogen.

Ergebnisse

Folgende Kategorien sind das Resultat des beschriebenen Vorgehens.

Relevanz des Codings im Unterricht:

Von der Mehrheit der Befragten wird das Coding als wichtigen Bestandteil des Informatikunterrichts betrachtet. Begründet wird dies mit Aussagen wie:

- *„Coding bietet Bezug zum logischen Denken“*
- *„Grundverständnis zum technischen Aufbau von Programmen ist wesentlich“*
- *„[...] wichtig für etwaige IT-Berufe oder IT-Studie“*
- *„um eine Vorstellung zu bekommen was hinter Anwendungen steckt“*
- *„Algorithmusverständnis ist wichtig – auch im Alltag“*

Allerdings wurde auch angemerkt, dass es auf den Detail-Grad und die Zielsetzung ankommt und in Relation zu den restlichen wichtigen Themen der Informatik gesehen werden muss. Zwei der Experten äußerten sich zu den vorherrschenden schlechten Office-Kenntnissen der Schülerinnen und Schüler und sehen hier einen größeren Handlungsbedarf. Wörtlich wurde dazu folgendes geäußert: *„Die Privatwirtschaft schreit nicht umsonst, dass beispielsweise die Office-Kenntnisse der SchulabgängerInnen teilweise desaströs sein sollen“* Auch erwähnten drei der Befragten, dass sie Ansätze, Coding bereits in der Volksschule zu unterrichten, für übertrieben halten. Dazu äußerte ein Experte: *„Die Ansätze des Codingsunterrichts in der Volksschule finde ich allerdings schlicht überzogen.“*

zeitliche Planung

Die Hälfte der Experten wünscht sich für dieses Konzept eine detaillierte Stundenplanung mit Minutenangaben bzw. eine Planungsmatrix. Weiters wurde angemerkt, dass bei den wenigen Einheiten welche im Laufe des Schuljahres zur Verfügung stehen nicht sehr viel Zeit für das Coding erübrigt werden könnte. Ein Experte meinte dazu: *„Die Zeitliche Planung wäre auch interessant [...]. Bedenken Sie, dass ein Schuljahr ca. 39 Wochen Unterricht ermöglicht, davon sind die SchülerInnen aber wahrscheinlich 5 nicht da (Fenstertage, Sommersportwoche, andere Ausflüge) In diesen restlichen 34 Doppelstunden muss der gesamte Lehrplan für das Jahr erfüllt werden.“* ein anderer schrieb: *„Coding ist sehr wichtig für SchülerInnen, muss allerdings in Relation zu all den anderen wichtigen Themen der Informatik gesehen werden. Das Schuljahr bringt nicht so viele Einheiten, dass man in der 5. Klasse das gesamte Schuljahr nur mit Coding verbringen könnte“.*

mögliche Probleme

Zwei Experten sahen in der Tatsache, dass der App Inventor nicht in deutscher Sprache zur Verfügung steht ein Problem. In diesem Zusammenhang meinte einer der Befragten: *„Sehe noch ein kleines Problem mit der englischen Sprache [...] SchülerInnen sind oftmals überraschend schlecht in Englisch und würden zu Hause bestimmt lieber (zumindest für die Entwicklungsumgebung) eine andere Sprache nutzen“*.

Die Verwendung eines Google-Accounts wurde ebenfalls von zwei Experten als problematisch betrachtet. In einem Fall wurde dies mit dem häufigen Vergessen von Login Daten begründet. Der zweite Experte begründete seine diesbezüglichen Vorbehalte nicht.

Einer der Befragten gab weiters zu bedenken, dass die Verwendung von Lernplattformen sich oftmals als schwierig erweist, da diese nicht immer online sind.

Erschwerend für die Umsetzung sehen zwei Experten das vorherrschende geringe Ausgangsniveau der Schülerinnen und Schüler. Ein Befragter meinten zum Beispiel: *„Auf der anderen Seite spüre ich in der Schule, dass Coding für Schüler mit 14 immer absolutes Neuland ist“* oder *„Das Niveau unser S ist nicht ausreichend, um aus Arbeitsblatt0 Informationen ziehen zu können“*.

Verantwortung und Vertrauen auf die Fähigkeiten der Schülerinnen und Schüler

Einer der Befragten ist der Meinung, dass die Verantwortung für die einzelnen Projektblöcke nur in „guten“ Klassen funktioniert und ein hoher Grad an Eigenverantwortung oft zu schlechten Ergebnissen führt. Wörtlich schrieb der Experte folgendes: *„Die Verantwortung während der einzelnen Projektblöcke hauptsächlich den Schülern zu übertragen kann in guten Klassen sehr gut funktionieren - in schlechteren Klassen führt so viel Eigenverantwortung oft zu schlechten Ergebnissen“*.

Ein andere Experte sieht vor allem in der Projektfindungsphase den Bedarf die Schülerinnen und Schüler zu lenken. Er meint dazu: *„Die Lehrenden sollten dennoch mehr in die Projektfindungsphase der Schülergruppen eingebunden werden, da man somit bereits am Anfang bei zu komplexen Themen einschreiten kann und nicht ein eventueller "Reset" hervorgerufen wird“*.

Weiters wurde vorgeschlagen komplexere Codeteile ohne Erklärung zum eintippen vorzugeben, um die Lernenden nicht zu überfordern.

Allerdings erwähnt einer der Befragten, welcher den App Inventor bereits im Unterricht eingesetzt hat, dass Anfangshürden schnell überwunden werden und die im Konzept enthaltenen Arbeitsblätter ein leichteres selbstständiges Arbeiten ermöglichen werden.

Themen im Unterricht neben dem Coding

Einmal wurde erwähnt, dass die Thematik des aktiven Zuhörens nicht gut in diesen Kontext passt. Der Experte meinte folgendes: *„Das Arbeitsblatt des "aktiven Zuhörens" ist meiner Meinung nach sehr gut gestaltet, aber mit dem vorliegenden Themenblock nicht unbedingt gut kombinierbar. Sollte man aber unbedingt für ein anderes Thema verwenden!“*

Angeregt wurde von einem Experten zusätzlich die Behandlung des Themas Web2.0 Applikationen, da diese sinnvoll und zeitgemäß sind.

Ein anderer Experte empfahl auch Lizenzen und Quellenangaben zu besprechen und meint: *„Vielleicht ist dieser Punkt (Lizenzen/CC) ebenfalls für die SchülerInnen interessant und sollte kurz thematisiert werden!“*

zu verbessern

Ein Experte empfiehlt, statt Arbeitsblatt0 eine Schritt-für-Schritt-Anleitung um den Schülerinnen und Schülern den Einstieg zu erleichtern.

Von einem der Befragten wurde vorgeschlagen, den Abschnitt „Aufbau des Konzepts“ durch eine Grafik zu ersetzen. Wörtlich meinte der Befragte: *„Die Seite 4 Aufbau des Unterrichtskonzepts hat mich etwas verwirrt. Ich würde die weglassen oder eine Grafik statt dessen einfügen“.*

Besonders gut

Ideen wie Reaktionsblätter, Projektideen-Verkauf, Tagesziele, Aufzeichnung wurden mehrfach als sehr positiv und interessant hervorgehoben. Die Experten äußerten unter anderem folgendes: *„Die Idee der Tagesziele, Reaktionsblätter sowie der "Projektideen-Verkauf" finde ich absolut großartig!!!“* oder *„Reaktionsblätter kannte ich bisher nicht, finde das aber sehr interessant“* und *„Besonders die Aufzeichnungen und die Gesprächskultur sind meiner Meinung nach wichtig.“*

Die Google-Account-Anleitung und Arbeitsblatt0 wurden ebenfalls von drei Experten als gut gelungen und hilfreich erachtet. Beispielsweise äußerte einer der Befragten: *„Besonderes Lob an die Idee mit der Google-Account Anleitung!!! Besonders hilfreich für ältere KollegInnen“.*

Auch wurde positiv erwähnt, dass der Überblick sehr detailliert beschrieben ist und die Schülerinnen und Schüler mithilfe der Arbeitsblätter gut geführt werden, aber trotzdem mehr selbst denken müssen als bei den auf der App Inventor Seite vorhandenen Tutorien. Auch wurde betont, dass diese ansprechend und altersadäquat gestaltet sind und ein rundes Gesamtbild ergeben. Dies wurde durch Aussagen wie: *„Bei dir werden sie wesentlich besser geführt, müssen aber mehr selbst denken (was ich super finde, denn bei den Tutorials steht alles Wort für Wort dort)“.* oder *„Super ausgearbeitet. Mit Videos und Bildern gut verständlich“* aber auch *„[...] Aufgaben passen auch wunderbar zur Altersgruppe“.* und *„Die Arbeitsblätter sind äußerst ansprechend und altersadäquat gestaltet und bilden ein rundes Gesamtes zum Schluss.“*

Positiv erwähnt wurde auch die Tatsache, dass die zur Bearbeitung der Arbeitsblätter benötigten Audio- und Bilddateien mit Quellenangaben bereitgestellt wurden.

das fehlt

Sowohl eine detaillierte zeitliche Planung als auch der Bezug zum Lehrplan wurde von einigen Experten vermisst. Geäußert wurde zum Beispiel: *„Konnte expliziten Lehrplanbezug nicht finden in den Unterlagen, aber super ist der Hinweis am Anfang auf mögliche Lernziele oder zu erreichende Kompetenzen“*. und *„Die Zeitliche Planung wäre auch interessant und der Bezug zum neuen Lehrplan [...]“* oder *„Bei der Detailansicht jedes Unterrichtsblocks hat mir irgendwie jeweils zu Beginn eine übersichtliche Planungsmatrix gefehlt.“*

Ein Experte hätte auch gerne einen Vergleich des App Inventors zu anderen blockbasierten Programmierumgebungen und die Frage, warum ausgerechnet der App Inventor verwendet werden soll, beantwortet. und meinte diesbezüglich: *„Mir fehlt eine Diskussion, in welcher der Benefit von App Inventor im Vergleich zu anderen vergleichbaren Plattformen deutlich wird. Zum Beispiel gibt für Schulen und LehrerInnen aktuell viele PocketCode sowie PocketPaint Fortbildungen und deren Anwendung hat bereits die Schulen in Ö erreicht. Aber es gibt auch weitere Konkurrenzprodukte, die keinen google account benötigen und somit vielleicht vorzuziehen wären. Es stellt sich also die Frage: Warum ausgerechnet der App Inventor?“*.

Die Beschreibung der Vorzüge des App Inventors, im Vergleich zu anderen blockbasierten Sprachen, ist bereits Bestandteil dieser Arbeit und wird dem Anhang des Unterrichtskonzepts beigelegt, bevor dieses veröffentlicht wird.

Empfohlen wurde außerdem ein Durchlauf des Konzepts, um häufige Fragen der Schülerinnen und Schüler in dieses aufnehmen zu können. Ein Befragter meinte: *„Es fehlt ein "Durchlauf" mit den Schülerinnen. Als Lehrperson mit Erfahrung in App Inventor "programmieren" ist für mich das Konzept klar, aber ich bin sicher, dass es Kleinigkeiten gibt die Schülerinnen in jedem Durchlauf immer wieder fragen werden. Dies wäre sinnvoll beim Arbeitsauftrag zu ergänzen, somit muss man nicht immer die gleichen Fragen individuell beantworten“*.

Ebenso wurde angeraten ein Programm für die Lehrerinnen- und Lehrerfortbildung zu entwickeln. Das wurde folgendermaßen zum Ausdruck gebracht: *„Großartig wäre aufbauend auf ihrem Konzept ein Programm für eine LehrerInnenfortbildung. Was müsste da gemacht werden? [...] Nicht alle unterrichten in einer AHS und haben ein INF Studium absolviert“*.

Conclusio und Ausblick

Es ist deutlich zu erkennen, dass Informatiklehrende mit Gerald Futschek, welcher meint, dass Coding ein wesentlicher Teil des Computational Thinkings und daher wichtig für die Schülerinnen und Schüler ist, übereinstimmen. (vgl. Gerald Futschek, 2016)

Es wird dabei jedoch auch angemerkt, dass die im Unterricht zur Verfügung stehende Zeit knapp bemessen ist und es neben dem Coding noch viele wichtige Themen in der Informatik gibt. Daher kann der Programmierunterricht nicht zu sehr ins Detail gehen. Es besteht die Hoffnung, dass sich die aktuelle Situation mit der Umsetzung des Projektes „Digitale Grundbildung“ (vgl. Bundesministerium für Bildung, Wissenschaft und Forschung) verbessert, was auch durch die Aussage eines Befragten zum Ausdruck kam.

Die Evaluation des Konzepts durch erfahrene Lehrpersonen sollte vor allem dazu beitragen dieses zu verbessern und Antworten auf folgende Fragen zu erhalten:

- *Ist ein derartiges Konzept geeignet, um sowohl die vom Lehrplan vorgegebenen informatischen, als auch sozialen Kompetenzen zu vermitteln und bei den Schülerinnen und Schülern Interesse an Programmierung und Projektmanagement zu wecken?*
- *Können sich Lehrende vorstellen die Rolle des Lernbegleiters einzunehmen?*
- *Sehen Lehrpersonen den sozialen Mehrwert den dieses Konzept zu vermitteln versucht?*

Zur Frage nach der Vermittlung der vom Lehrplan vorgegebenen informatischen und sozialen Kompetenzen fanden die Befragten das Konzept geeignet und die Beispiele wurden als altersgerecht eingestuft. Bei der Erstellung des Unterrichtskonzeptes wurde darauf geachtet, dass die Aufgaben auf das Lösen realer Probleme ausgelegt sind, Eigeninitiative und selbstgesteuertes Lernen ermöglicht wird und zwischenmenschliche Beziehungen, sowie Kommunikationsfähigkeiten gefördert werden. Diese Kombination soll dazu beitragen signifikantes Lernen zu ermöglichen und das Interesse der Lernenden an Programmierung und Projektmanagement zu wecken. Ob dies tatsächlich mit diesem Konzept verwirklicht wird, kann erst die konkrete Umsetzung durch eine Lehrperson und Klasse ergeben. Neben einem Konzept ist dafür das Herstellen eines lernförderlichen Klimas im Sinne des personenzentrierten Ansatzes unerlässlich. Dieses Klima muss durch die Lernenden auch wahrgenommen werden, wozu es der eigenen Erfahrung bedarf.

Die Antwort auf die zweite Frage ist differenziert. Manche Lehrende sehen sich bereits jetzt primär als Lernbegleiter. Andere meinen, sowohl als Lernbegleiter als auch als Instruktor wirken zu wollen. Im traditionellen Schulsystem ist es vermutlich kaum möglich, ausschließlich die Funktion eines Lernbegleiters zu übernehmen, da die vom Lehrplan vorgegebenen fachlichen Ziele umzusetzen sind. Auch mit diesem Konzept wird eine Mischform zwischen Instruktor und Lernbegleiter anvisiert. Die

Lernziele werden anhand der bereitgestellten Unterlagen vorgegeben, was dem Lehrenden ermöglichen soll, während der Einheiten überwiegend als Begleiter zu agieren.

Einige Äußerungen bei den offenen Fragen, lassen darauf schließen, dass das Vertrauen in die Schülerinnen und Schüler, selbstständig aktiv zu werden nicht immer vorhanden ist. Auch der Wunsch nach einer detaillierten Zeitplanung widerspricht meines Erachtens den Prinzipien des personenzentrierten Unterrichts, da dieser, wie von Renate Motschnig-Pitrik und Michael Derntl festgestellt wurde, auch Raum für Unerwartetes und Flexibilität von Lehrenden und Lernenden voraussetzt. (vgl. Motschnig-Pitrik & Derntl, Michael, 2002)

Trotz der bestehenden Bedenken der Lehrenden in Bezug auf die Eigenverantwortlichkeit der Schülerinnen und Schüler, kann sich nach Durchsicht des vorgestellten Konzeptes die Hälfte der Befragten vorstellen die Funktion des Lernbegleiters einzunehmen.

Obwohl die im Konzept vorgesehenen Gruppengespräche, Präsentationen und Reaktionsblätter überwiegend als gute Ergänzung gesehen werden, wurde der soziale Mehrwert nur teilweise erkannt. Eine mögliche Interpretation dieses Ergebnisses wäre, dass die befragten Lehrpersonen, welche nach eigenen Angaben die Vermittlung von Soft-Skills im Unterricht für essenziell erachten, stets bemüht sind soziale Kompetenzen zu vermitteln und somit dieses Konzept zwar ihren Standard erfüllt aber nur bedingten Mehrwert in Bezug darauf erzeugt. Auf Nachfrage bei den Experten, meinte einer dazu: *„Ich denke, dass die soziale Komponente weniger relevant ist, weil die S inhaltlich sehr gefordert sein werden und es wichtig ist, dass jeder für sich etwas leistet.“*

Das Unterrichtskonzept wird aufgrund der Expertenbefragung um die Bezüge zum Lehrplan der AHS-Oberstufe ergänzt, da dies einer der Punkte ist, der mehrfach urgiert wird. Dem ebenfalls mehrfach zum Ausdruck gebrachten Wunsch nach einer detaillierten Zeitplanung wird aus den oben angeführten Gründen und um das Unterrichtskonzept flexibel halten zu können, da es sowohl für das Unterrichtsfach Informatik als auch für das Wahlfach Informatik geeignet sein soll, nicht entsprochen. Es soll zudem auf die jeweilige Ausgangssituation (Gruppengröße, Vorkenntnisse) angepasst werden können.

Drei Experten werden das Konzept in ihrem Unterricht einsetzen und mir anschließend ihre Erfahrungen, sowie die daraus resultierenden Verbesserungsvorschläge zukommen lassen. An einem dieser Durchläufe werde ich persönlich teilnehmen, um häufig auftretende Fragen der Schülerinnen und Schüler zu dokumentieren und die dazugehörigen Antworten im Unterrichtskonzept aufzunehmen.

ZUSAMMENFASSUNG

In der vorliegenden Diplomarbeit steht die Bedeutung der Kommunikationsfähigkeiten in der Informatik, der personenzentrierte Unterricht nach Carl R. Rogers, sowie dessen mögliche Umsetzung im Informatikunterricht im Vordergrund.

Im ersten Teil der Arbeit wird, nach ausführlicher Literaturrecherche ein Überblick über die Relevanz von Kommunikationsfähigkeiten erläutert und dargestellt, inwiefern im Lehrplan der AHS darauf Bezug genommen wird. Ebenso werden der personenzentrierte Ansatz und die humanistische Pädagogik thematisiert. Im Detail wird hier auf den personenzentrierten Unterricht, signifikantes Lernen und aktives Zuhören aber auch auf die zur Umsetzung dieser Theorien notwendige Haltung der Lehrenden und Lernenden eingegangen.

Im weiteren Verlauf wird das Konzept „In-ClassFlip“, sowie dessen Bezug zu lerntheoretischen Ansätzen dargestellt und die Vor- und Nachteile visueller Programmierung und im Speziellen des App Inventors beleuchtet, sowie das Thema Computational Thinking bearbeitet.

Um all diese Theorien zu vereinen, wird im praktischen Teil dieser Arbeit ein Unterrichtskonzept zur „Einführung in die Programmierung mittels App-Inventor“ vorgestellt, welches Lehrende bei der Umsetzung personenzentrierten Unterrichts unterstützen soll. Dieses Konzept wird erfahrenen Lehrenden zur Begutachtung vorgelegt. Das Feedback aus der anschließenden Expertenbefragung wird in weiterer Folge analysiert und reflektiert.

ABSTRACT

The present paper is focused on one hand on communication skills in computer science, on the other hand on person-centered classes, adapted from Carl R. Rogers, as well as on the possibilities to implement those in class.

The first part of the thesis will give an overview of the impact of communication skills and how the curriculum of the “AHS” is relating to it. Furthermore the person-centered approach and humanistic educational science will be discussed. In detail it is about significant learning and active listening, but also about the necessary position of teachers to implement those theories.

Additionally the concept of “In-ClassFlip” and the relation to theoretical study approach will be discussed. The focus is on pros and cons of visual programming, in particular on the view of the App Inventor, as well as the topic computational thinking.

To combine all these theories the practical part of the paper will show a teaching concept about the “Introduction in programming via App-Inventor”. This concept shall be useful for teachers to implement person-centered classes. The idea was given to experienced teachers to examine. The feedback of the survey will be analysed and reflected in further consequence.

ABBILDUNGSVERZEICHNIS

Abbildung 1 - Auswertung Aufbau und Struktur	68
Abbildung 2 - Auswertung Beispielaufgaben	68
Abbildung 3 - Auswertung App Inventor zur Einführung in die Programmierung	69
Abbildung 4 - Auswertung Erfahrungen mit dem App Inventor	69
Abbildung 5 - Auswertung Vermittlung sozialer Fähigkeiten	70
Abbildung 6 - Auswertung Verwendbarkeit	70
Abbildung 7 - Auswertung Ergänzungen.....	70
Abbildung 8 - Auswertung Rolle der Lehrperson	71

LITERATURVERZEICHNIS

- Abelson Hal. (2009). App Inventor for Android. Retrieved February 15, 2018, from <https://research.googleblog.com/2009/07/app-inventor-for-android.html>
- About Us | Explore MIT App Inventor. Retrieved April 16, 2018, from <http://appinventor.mit.edu/explore/about-us.html>
- Algodoo.. Retrieved April 13, 2018, from <http://www.algodoo.com/>
- Anderson, Lorin W. u. a. (2001). *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*. London: Longman Publishing Group.
- App Inventor Classic | Explore MIT App Inventor. Retrieved January 15, 2018, from <http://appinventor.mit.edu/explore/classic.html>
- Bandura A. (2001). Social cognitive theory: An agentic perspective. *Annual Review of Psychology*, pp. 1–26.
- Barr, V., Stephenson, C. (2011). *Bringing computational thinking to K-12*. (2nd ed.). ACM Inroads.
- Barrett-Lennard, Godfrey T. (1998). *Carl Rogers' Helping System: Journey & Substance*. SAGE.
- Bau David, Gray Jeff, Kelleher Caitlin, Sheldon Josh, & Turbak Franklyn. (2017). Learnable Programming: Blocks and Beyond. *Communications of the ACM*, 72–80.
- Bay, R.H. (1998). *Erfolgreiche Gespräche durch aktives Zuhören* (3rd ed.). Renningen-Malmsheim: Expert Verlag.
- Bennett, Brian E., Spencer, Dan, Bergmann, Jon, Cockrum, Troy, Musallam, Ramsey, Sams, Aaron, Overmyer, Jerry. (2013). The Flipped Class Manifest - THE DAILY RIFF - Be Smarter. About Education. Retrieved December 16, 2017, from <http://www.thedailyriff.com/articles/the-flipped-class-manifest-823.php>
- Bergmann ; Jonathan, & Sams, Aaron. (2012). *Flip your Classroom - reach Every Student in Every Class Every Day*. Library of Congress Cataloging-in-Publication Data.
- BMB. Verbindliche Übung „Digitale Grundbildung“ in Sekundarstufe 1. Inhalte für Pilotierung im Schuljahr 2017/18, 8.
- Bortz, Jürgen, & Döring, Nicola. (2007). *Forschungsmethoden und Evaluation für Human- und Sozialwissenschaftler: Limitierte Sonderausgabe* (4th ed.). Springer-Verlag.

Bouley B. du. (1989). Some difficulties of learning to program. *Soloway E. (Hrsg.); Sphorer, J.C. (Hrsg.) Studying Th Novice Programmer*, pp. 283–299.

Brei Jennifer, & Dr. Wagner Eike. Soft Skills im Projektmanagement. *PMI Munich Chapter e.V.*
Retrieved January 03, 2018 from
https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&cad=rja&uact=8&ved=2ahUKEwid8LrUvpbbAhUE-qQKHbpyAjsQFjACegQIARA1&url=http%3A%2F%2Fs11729d18e8ad7bb7.jimcontent.com%2Fdownload%2Fversion%2F1364313495%2Fmodule%2F7335600286%2Fname%2FSoft%2520Skills%2520in%2520der%2520Projektarbeit_Wagner%2520Brei.pdf&usg=AOvVaw0JSt8A5lVv46UCLm-qFNOR

Brennan K. and Resnick M. (2012). New frameworks for studying and assessing the development of computational thinking. Proceedings of the 2012 annual meeting of the American Educational Research Association, Vancouver, Canada.

Bruner J., Olver R., & Greenfield, P.M. (1988). *Über kognitive Entwicklung* (2nd ed., Vol. Studien zur kognitiven Entwicklung). Klett-Cotta.

Bugental, James F. T. (1964). The Third Force in Psychology". *Journal of Humanistic Psychology* 4, pp. 19–25.

Bundesministerium für Bildung, Wissenschaft und Forschung.. BMBWF :Digitale Grundbildung.
Retrieved March 21, 2018, from <https://bildung.bmbwf.gv.at/schulen/schule40/dgb/index.html>

Bundy A. (2007). *Computational thinking is pervasive* (Vol.1, No.2 p. 67-69). Journal of Scientific an Practical Computing.

Clark Andrew. (2013, December 30). App Inventor launches second iteration. Retrieved April 15, 2018, from <http://news.mit.edu/2013/app-inventor-launches-second-iteration>

Cornelius-White; J. (2007). Learner-centered teacher–student relationships are effective: a meta-analysis. *Rev. Educ. Res.* 77, pp. 1–31.

Damasio, A. (2000). The Feeling of What Happens - Body, Emotion and the Making of Consciousness. *Vintage*.

Denning P. J. (2009). The profession of IT beyond computational thinkng. *Communications of the ACM*, pp. 28–30.

Donald E. Knuth Lectures. (2014, July 8). Retrieved February 17, 2018, from
<http://scpdweb.stanford.edu/free-stuff/engineering-archives/donald-e-knuth-lectures>

Dumke, Reiner (2013). *Software Engineering: Eine Einführung für Informatiker und Ingenieure* (4th ed.). Springer Verlag.

- Flick, Uwe. (2006). *Qualitative Evaluationsforschung; Konzepte Methoden Umsetzungen*. Reinbek bei Hamburg: Rowohlt Verlag GmbH.
- Freitag, M. (2016). *Kommunikation im Projektmanagement: Aufgabenfelder und Funktionen der Projektkommunikation*. Springer-Verlag.
- Futschek, Gerald (2016). Computational Thinking im Unterricht. Schule Aktiv. Retrieved February 02, 2018 from https://publik.tuwien.ac.at/files/PubDat_251743.pdf;
- GameSalad – Empowering Game Creators. Retrieved April 13, 2018, from <https://gamesalad.com/>
- Geipel, P. (2003). *Der IT-Projektmanager: Arbeitstechniken, Checklisten und soziale Kompetenz*. Pearson Deutschland GmbH.
- Gindling Jim, Ioannidou Andri, Loh Jennifer, Lokkebo Olav, & Repenning Alexander. (1995). LEGOsheet: A Rule-Based Programming, Simulation and Manipulation Environment for the LEGO Programmable Brick, 172–179.
- Groddeck, N. (2006). *Carl Rogers. Wegbereiter der modernen Psychotherapie* (2nd ed.). Darmstadt: Wissenschaftliche Buchgesellschaft.
- Hartmann Werner, Näf Michael, & Reichert Raimond. (2006). *Informatikunterricht planen und durchführen*. Springer Verlag.
- Hinte, W. (1980). *Non-direktive Pädagogik; Eine Einführung in Grundlagen und Praxis des selbstbestimmten Lernens* (Vol. vol 41.). VS Verlag für Sozialwissenschaften.
- Isaacs, W (2002); *Dialogue and the art of thinking together; Deutsche Version Isaacs, W (2002) Dialog als Kunst Gemeinsam zu Denken*. (1999). Bergisch Gladbach: EHP - Edition Humanistische Psychologie.
- ISTE: computational-thinking-operational-definition-flyer.pdf. Retrieved January 22, 2018 from <http://www.iste.org/docs/ct-documents/computational-thinking-operational-definition-flyer.pdf>;
- Jong de; Ton. (2009, March). Cognitive load theory, educational research, and instructional design: some food for thought. *Springerlink.Com*. Retrieved February 17, 2018 from <https://link.springer.com/content/pdf/10.1007/s11251-009-9110-0.pdf>;
- Kay, Alan C. (1993). The Early History Of Smalltalk. Retrieved January 13, 2018, from <http://propella.sakura.ne.jp/earlyHistoryST/EarlyHistoryST.html>
- Keil Wolfgang W., & Stumm Gerhard. (2013). *Die vielen Gesichter der personenzentrierten Psychotheapie*. Springer Verlag.

- Kelleher Caitlin, & Pausch Randy. (2005). Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM Computing Surveys*, pp. 83–137.
- Kent-Noss; EE2002-preprint.pdf. Retrieved January 03, 2018 from <http://www.oemg.ac.at/FH/Klagenfurt2005/Kent-Noss-EE2002-preprint.pdf>
- Kim, M.K., Kim, S.M., Khera, O., & Getman, J. (2014). *The experience of three flipped classrooms in an urban university: an exploration of design principles. Internet and Higher Education* (22nd ed.). The Internet an Higher Education.
- Knuth; Donald E. (1997). *The Art of computer programming* (3rd ed., Vol. 1). Stanford University: Addison-Wesley.
- Landerer Claudio. (2006). *Die Nintendo-Generation lernt programmieren*. Universität Salzburg.
- Lang; Sabine. Empirische Forschungsmethoden. Universität Trier. Retrieved Mai 01, 2018 from https://www.uni-trier.de/fileadmin/fb1/prof/PAD/SP2/Allgemein/Lang_Skript_komplett.pdf; Logo History. Retrieved April 12, 2018, from http://el.media.mit.edu/logo-foundation/what_is_logo/history.html
- Lutz Berger, Joachim Grzega, Christian Spannagel. (2009). *Lernen durch Lehren im Fokus*.
- Mager, R.F. (1972). *Lernziele und programmierter Unterricht* (35th ed.). Weinheim: Beltz.
- Mayring; Philipp. (2010). *Qualitative Inhaltsanalyse In: Mey G., Mruck K. (eds): Handbuch Qualitative Forschung in der Psychologie p. 601-613*. Wiesbaden: VS Verlag für Sozialwissenschaften | Springer Fachmedien. Retrieved Mai 04, 2018 from <https://doi-org.uaccess.univie.ac.at/10.1007/978-3-531-92052-8>
- Miller G. A. (1956). The magical number seven, plus or minus two, some limits on our capacity for processing information. *Psychological Review*, pp. 81–97.
- Mayer, Richard E. (2009); *Multimedia Learning, Second Edition*; Cambridge University Press; New York; First Edition published 2001;
- Mayer, Richard E. & Roxana Moreno, (1998); *A Split-Attention Effect in Multimedia Learning: Evidence für Dual Processing Systems in Working Memory*; Univerity of California; Santa Barbara; Journal of Educational Psychology Vol. 90; No.2; p 312-320 American Psychological Association, Inc. Retrieved February 24, 2018 from <http://visuallearningresearch.wiki.educ.msu.edu/file/view/Mayer+%26+Moreno+%281998%29.pdf>
- Motschnig Renate. (2010). Informatikunterricht vermittelt vielfältige Kompetenzen. Können wir Lernende als “ganzes Wesen” erreichen. *25 Jahre Schulinformatik in Österreich G. Brandhofer, G. Futschek, P. Micheuz, A. Reiter Und K.Schoder (Hrsg.)*, pp. 280–282.

- Motschnig, Renate, & Nykl Ladislav. (2003). Toward a cognitive-emotional model of Roger's person-centered approach. *Journal of Humanistic Psychology* 43 No.4, pp. 8–45.
- Motschnig Renate, & Nykl Ladislav. (2013). *Konstruktive Kommunikation Sich und andere verstehen durch personenzentrierte Interaktion*. Klett-Cotta.
- Motschnig Renate, & Nykl Ladislav. Der Personenzentrierte Ansatz nach Carl R. Rogers. "Person-Centred Approach" *Psychotherapie, Leben- Und Sozialberatung, MMag. Ladislav Nykl PhD*. Retrieved February 09, 2018 from <https://cewebs.cs.univie.ac.at/pm-ss/ws04/Files/Exkurs%20zum%20personenzentrierten%20Ansatz.pdf>
- Motschnig, Renate, Sedlmair, Michael, Schroder, Svenja, & Möller, Torsten. (2016, October). A team-approach to putting learner-centered principles to practice in a large course on Human-Computer Interaction. *Frontiers in Education Conference (FIE), 2016 IEEE*. Retrieved December 15, 2017 from https://eprints.cs.univie.ac.at/4765/1/Team-Approach_160707.pdf
- Motschnig-Pitrik, R. (2005). Person-Centered e-Learning in Action: Can Technology Help to Manifest Person-Centered Values in Academic Environments? *Journal of Humanistic Psychology* 45, pp. 503–530.
- Motschnig-Pitrik, R., & Derntl, Michael. (2002). Student-Centered eLearning (SCeL): Concept and application in a students' project on supporting learning. *Department of Computer Science and Business Informatics*. Retrieved January 13, 2018 from <https://eprints.cs.univie.ac.at/1122/1/SCeL02.pdf>
- Motschnig-Pitrik, R. (2004). Blended Learning in einer großen Informatik-Lehrveranstaltung: Personenzentriert oder Handlungsorientiert? *Handlungsorientiertes Lernen Und ELearning: Grundlagen Und Praxisbeispiele*, pp. 219–246.
- Motschnig-Pitrik, R. & Mallich, K. (2004). Effects of Person-Centered Attitudes on Professional and Social Competence in a Blended Learning Paradigm. *Educational Technology & Society*, pp. 176–192.
- Motschnig-Pitrik, R. & Mallich, Katherina. (2002). Learning at three levels: Web-support and experiential learning in project management. *Proceedings of the International Workshop on Interactive ComputerAided Learning*.
- Motschnig-Pitrik, Renate, & Derntl, Michael. (2008). Three scenarios on enhancing learning by providing universal access. *Univ Access Inf Soc (2008) 7: 247*. <https://doi-org.uaccess.univie.ac.at/10.1007/S10209-008-0121-4>.
- Motschnig-Pitrik, Renate, (2006). Technology Enhanced Learning auf Basis des Personenzentrierten Ansatzes. *ELearning an Der Universität Wien; Waxmann Verlag*.
- Open Education. (2016, December 13). In *Wikipedia*. Retrieved April 04, 2018 from https://de.wikipedia.org/w/index.php?title=Open_Education&oldid=160610900

- Paivio, A. (1986). *Mental representations: A dual-coding approach*. Oxford University Press.
- Pant, Ira, & Baroudi, Bassam. (2006, March 23). Projekt management education: The human skills imperative. School of Natural and Built Environments, University of South Australia (City East), North Terrace, Adelaide, South Australia. Retrieved February 12, 2018 from <http://isiarticles.com/bundles/Article/pre/pdf/3199.pdf>
- Papert Seymour. (1999). *Logo Philosophy an Implementation*. Logo Computer Systems Inc.
- Papert Seymour, & Harel, Idit. (1991). Situatoin Constructionism. *First Chaper of Constructionism in Ablex Publishing Corporation*. Retrieved December 29, 2018 from <http://www.papert.org/articles/SituatingConstructionism.html>
- Peppler K., & Kafai Y. (2007, July). From SuperGoo to Scratch: Exploring creative media production in informal learning. *Jounal on Learning, Media, and Technology*, pp. 149–166.
- Porst; Rolf. (2014). *Fragebogen; (Hrsg.) Sahner H., Halle, Bayer M., Nürnberg, Sackmann, R. Halle* (4th ed.). Wiesbaden: Springer VS.
- Pörtner, M. (2008). *Ernstnehmen - Zutrauen - Verstehen; personzentrierte Haltung im Umgang mit geistig behinderten und pflegebedürftigen Menschen*. Stuttgart: Klett-Cotta.
- Price Thomas W., & Barnes Tiffany. (2015). Comparing Textual and Block Interfaces in a Novice Programming Environment. *ACM*.
- Resnick; Michael. (1996, July). Distributed Constructionism. *Proceedings of the International Conference on the Learning Sciences Association for the Advancement of Computing in Education*. Retrieved January 06, 2018 from <https://llk.media.mit.edu/papers/archive/Distrib-Construct.html>
- Resnik M., Meloney J., Monroy-Hernández, Rusk N., Eastmond E., Brennan K., Kafai Y. (2009). Scratch: programming for all. *Comm. ACM*, pp. 60–67.
- RIS - Lehrpläne – allgemeinbildende höhere Schulen - Bundesrecht konsolidiert, Fassung vom 18.03.2018. Retrieved March 18, 2018, from <https://www.ris.bka.gv.at/GeltendeFassung.wxe?Abfrage=Bundesnormen&Gesetzesnummer=10008568&FassungVom=2017-08-31>
- RIS - Schulorganisationsgesetz § 2 - Bundesrecht konsolidiert. Retrieved March 19, 2018, from <https://www.ris.bka.gv.at/Dokument.wxe?Abfrage=Bundesnormen%20&Dokumentnummer=NR012118405>
- Rogers, Carl R., C. R. (1983/2005). *Die klientenzentrierte Gesprächspsychotherapie - Client-Centered Therapy*. Frankfurt am Main: Fischer Verlag.

- Rogers; Carl Ransom, C. R. (1969). *Freedom to Learn*. Columbus, Ohio: Charles E. Merrill Publishing Company.
- Rogers, C.R. (1951/1995). *Client-centered-therapy*. London: Constable.
- Rogers, C.R. (1959/1991). Eine Theorie der Psychotherapie, der Persönlichkeit und der zwischenmenschlichen Beziehungen" GwG Verlag, Köln, 3. Auflage; Original: A theory of therapy, personality, and interpersonal relationship, as developed in the client-centered framework. *S. Koch Psychology. A Study of a Science, Vo.III*, 184–256.
- Rogers, C.R. (1961/1985). *On Becoming A Person - A Psychotherapists View of Psychotherapy*. London: Constable.
- Rogers, C.R. (1942/1972). *Die nicht-direktive Beraterung* (2nd ed.). München: Kindler-TB. Original: (1942) Counseling and psychotherapy. Boston: Houghton Mifflin.
- Rogers, C.R. (1973/2009). *Entwicklung der Persönlichkeit* (17th ed.). Stuttgart: Klett-Cotta.
- Rogers, C.R. (1980). *A way of being*. Boston: Houghton and Mifflin Company.
- Rogers, C.R. (1988). *Lernen in Freiheit. Zur Bildungsreform in Schule und Universität*. Frankfurt am Main: Fischer Verlag.
- Rogers, C.R., & Farson, R.E. (1957). *Aktive Listening*. Chicago: University of Chicago Industrial Relations Center.
- Rogers, C.R., & Rosenberg Rachel L. (1980). *Die Person als Mittelpunkt der Wirklichkeit* (2nd ed.). Stuttgart: Klett-Cotta.
- Schiffer, Stephan. (1998). *Visuelle Programmierung: Grundlagen und Einsatzmöglichkeiten*. Addison-Wesley-Longman.
- Schulz von Thun, F. (1981). *Miteinander reden 1 - Störungen und Klärungen: Allgemeine Psychologie der Kommunikation*. Reinbek bei Hamburg: Rowohlt Taschenbuch Verlag GmbH.
- Sürig, Nicole (2006). *Die Pro- und Contradebatte als handlungsorientierte Unterrichtsmethode im Fach Sozialkunde*.
- SoSci Survey. Retrieved May 18, 2018, from <https://www.soscisurvey.de/>
- Spannagel; Christian. (2013). *Open Education und Open Science*. Retrieved April 11, 2018 from <https://www.youtube.com/watch?v=PDweerwQPVw>
- Spannagel; Christian. *10 Irrtümer zum Einsatz digitaler Medien in der Schule*. Oliver Tracke. Retrieved March 03, 2018 from <https://www.youtube.com/watch?v=HsXP528OVtU&feature=youtu.be>

- Spencer, Dan, Wolf, Deb, & Sams, Aaron. (2011). Are You Ready to Flip? - THE DAILY RIFF - Be Smarter. About Education. Retrieved March 24, 2018, from <http://www.thedailyriff.com/articles/are-you-ready-to-flip-691.php>
- Stagecast Creator. Retrieved April 13, 2018, from <http://acypher.com/creator/>
- Stevens Matt, & Norman Richard. (2016). Industry Expectations of Soft Skills in IT Graduates. *ACE*, (16).
- Sweller, John, Merrienboer van; Jeroen J.G., & Paas, Fred G.W.C. (1998). Cognitive Architecture and Instructional Design. *Educational Psychology Review*.
- Tausch, Reinhard, & Tausch, Anne-Marie. (1991). *Erziehungspsychologie. Begegnung von Person zu Person* (10th ed.). Göttingen: Hogrefe.
- Think-pair-share. (2017, May 6). In *Wikipedia*. Retrieved February 07, 2018 from <https://en.wikipedia.org/w/index.php?title=Think-pair-share&oldid=778951164>
- Tiago Miranda, & Bimal Ghimire. (2007). Desired Competences for Project Managers. Retrieved January 12, 2018 from <http://www.diva-portal.org/smash/get/diva2:141277/FULLTEXT01.pdf>
- Vier-Seiten-Modell. (2018, January 4). In *Wikipedia*. Retrieved January 04, 2018 from <https://de.wikipedia.org/w/index.php?title=Vier-Seiten-Modell&oldid=172574396>
- Watzlawick; Paul. (2017, December 31). In *Wikipedia*. Retrieved Mai 13, 2018 from https://de.wikipedia.org/w/index.php?title=Paul_Watzlawick&oldid=172456921
- Weinert; Franz E. (Hrsg). (2001). *Vergleichende Leistungsmessungen in Schulen*. Weinheim und Basel: Beltz.
- Weintrop David, & Wilensky Uri. (2017). How Block-based Languages Support Novices: A Framework for Categorizing Block-based Affordances, 92–99.
- Who We Are. Retrieved March 30, 2018, from <https://flippedlearning.org/who-we-are/>
- Wing; Jeannette M. (2014). Computational Thinking Benefits Society |. Retrieved March 19, 2018, from <http://socialissues.cs.toronto.edu/index.html%3Fp=279.html>
- Wing J.M. (2006). Computational thinking. *Communications of The ACM*, 49/3, pp. 33–35.

ANHANG

Einführung in die Programmierung mit App Inventor



INHALTSVERZEICHNIS

Vorwort.....	4
Aufbau dieses Unterrichtskonzepts	5
Motivation den App-Inventor im Unterricht einzusetzen.....	6
Grundsätzliches zum App Inventor	6
Zielgruppe, Lernziele und Kompetenzen	7
Zielgruppe	7
Lernziele und Kompetenzen	7
Bezug zum Lehrplan Informatik der AHS.....	7
Voraussetzungen	9
Voraussetzungen auf Seiten der Lehrperson	9
Voraussetzungen auf Seiten der Schüler und Schülerinnen	9
Ein möglicher Unterrichtsablauf.....	10
1. Block	10
2. Block.....	10
3. Block	11
4. Block	11
5. Block	12
6. Block	12
Ergebnissicherung:.....	12
Einstieg in den App Inventor 2	13
Systemanforderungen:	13
Vorbereitung und Installation:.....	13
Möglichkeit 1 (bevorzugte Möglichkeit)	14
Möglichkeit 2.....	15
Möglichkeit 3.....	15
Apps verpacken und teilen:.....	16
Der App Inventor im Überblick	17
Arbeitsblatt 0	22
Die Lernplattform	23
Arbeitsblätter	23
Videos und Internetlinks	23
Reaktionsblätter	23
Anhang	24
Arbeitsblatt aktives Zuhören	24

Arbeitsblatt 1 und Arbeitsblatt 2	25
Arbeitsblatt 3.....	30
Arbeitsblatt 4 – MOLE MASH.....	35
Arbeitsblatt 5 – PONG	46
Videos erstellen	54
Google Account.....	56
Quellen und verwendete Literatur	57
Abbildungsverzeichnis	57
Quellen:	57
Bildnachweise:	57
weiterführende Literatur:	58

Vorwort

Liebe Kollegen und Kolleginnen!

Mit diesem Konzept, welches im Rahmen meiner Diplomarbeit verfasst wurde, würde ich gerne Lehrer und Lehrerinnen motivieren, gemäß den Theorien von Carl Rogers, ihre Rolle im Klassenzimmer vom Experten und Fachlehrer hin zum Lerndesigner, Lernbegleiter, Mitlernenden und Unterstützer zu verändern. Im Fokus dieser Arbeit steht die Art und Weise wie signifikantes Lernen bei Schülerinnen und Schülern initiiert werden kann und gleichzeitig Kommunikationsfähigkeiten trainiert werden können.

Es soll eine Möglichkeit aufzeigen, die „Einführung in die Programmierung“ und die Vermittlung sozialer Kompetenzen zu kombinieren. Es ist ebenfalls als Hilfestellung und Motivation für diejenigen gedacht, welche noch keine oder wenig Erfahrungen mit dem App Inventor gemacht haben, welcher sich aufgrund seiner grafischen Oberfläche gut eignet, einfache Anwendungen für mobile Endgeräte mit dem Betriebssystem Android zu erstellen.

Dieses Konzept beinhaltet Unterlagen, um ein komplettes Modul zur prozeduralen Programmierung mithilfe des App Inventors aufzubauen. Der Konnex zur Lebenswelt der Schüler und Schülerinnen ist dabei durch den Stellenwert des Smartphones im Alltag und den Umstand, dass Spiele, welche bei vielen bekannt sind, programmiert werden, gegeben. Die erworbenen Grundkenntnisse lassen sich dann später auf andere höhere Programmiersprachen übertragen.

Zu Beginn werden die Schüler und Schülerinnen anhand von einfachen Beispielen mit dem App Inventor vertraut gemacht. Die weiteren Arbeitsblätter unterstützen sie bei der Erstellung eigener Spiele Apps und enthalten Links zu allen verwendeten Medien, um ein zügiges Vorankommen zu ermöglichen. Die Schüler und Schülerinnen können im weiteren Verlauf ihre Spiele nach eigenen Vorstellungen verbessern und ausbauen oder ein eigenes Projekt umsetzen. Spielerisches und entdeckendes Lernen soll mithilfe der Unterlagen ermöglicht werden.

Ähnlich wie beim „Flipped Classroom“, bei dem Erklärungen und Vorträge mithilfe von Videos und Arbeitsblättern in die Vorbereitungszeit der Schüler und Schülerinnen verlagert werden, ist es bei diesem Konzept vorgesehen, dass eine weitestgehend selbstständige Wissensaneignung anhand der zur Verfügung gestellten Unterlagen erfolgt.

Ergänzend zur Bearbeitung der bereitgestellten Unterlagen sind regelmäßige Gesprächsrunden, Präsentationen und Reaktionsblätter geplant.

Die zentralen Themen, visuelle Programmierung und computational thinking im Unterricht, signifikantes Lernen, aktives Zuhören und personenzentrierte Kommunikation sowie Soft-Skills in der Informatik, werden in der begleitenden Diplomarbeit genauer behandelt.

Aufbau dieses Unterrichtskonzepts

Motivation den App Inventor im Unterricht einzusetzen

Mögliche Vorteile, welche der Einsatz des App Inventors im Unterricht bringt.

Grundsätzliches zum App Inventor

Was ist der App Inventor? Was wird zum Arbeiten mit dem App Inventor benötigt?

Zielgruppe, Lernziele und Kompetenzen & Voraussetzungen

Für wen ist dieses Konzept gedacht, welche Lehr- und Lernziele deckt dieses Konzept ab und welche Voraussetzungen sollten die Lehrperson und die Schüler und Schülerinnen mitbringen?

Ein möglicher Unterrichtsverlauf & Ergebnissicherung

Es wird eine Idee eines möglichen Verlaufs für ein Modul „Einführung in die Programmierung“ präsentiert. Dieser Entwurf beinhaltet, neben der eigenständigen Erarbeitung der zur Realisierung des vorgesehenen Projekts erforderlichen Programmierkenntnisse, verschiedene Kommunikationsphasen wie Präsentationen, Feedback und Gruppengespräche, da Kommunikationsfähigkeiten als essenziell für die Informatik und speziell für das Projektmanagement angesehen werden.

Wie kann festgestellt werden, ob die Schüler und Schülerinnen die vom Lehrplan geforderten Kompetenzen erworben haben?

Einstieg in der App Inventor & Der App Inventor im Überblick

Dieser Abschnitt gibt einen kurzen Überblick über die Systemanforderungen und zum Testen benötigten Installationen sowie über die Möglichkeiten erstellte Anwendungen zu sichern und zu teilen.

Es wird die Programmierumgebung beschrieben, um der Lehrperson einen raschen Überblick zu verschaffen. Die wichtigsten Informationen wurden auch auf dem Arbeitsblatt_0 zusammengefasst, welches den Schülern und Schülerinnen bereitgestellt werden kann.

Die Lernplattform

Liefert eine kurze Beschreibung welche Materialien und Möglichkeiten auf eine Lernplattform gestellt werden sollen.

Anhang

Im Anhang befinden sich die Arbeitsblätter, welche aktives und signifikantes Lernen fördern sollen, Tipps zur Erstellung eigener Erklärvideos und ein Arbeitsblatt zum Einrichten und Löschen eines Google-Accounts.

Quellen und verwendete Literatur

Im Unterrichtskonzept wurde teilweise auf die Erwähnung der Quellen verzichtet, um eine bessere Lesbarkeit zu gewährleisten. Diese und Hinweise auf weiterführende Literatur werden in diesem Abschnitt angegeben.

Motivation den App-Inventor im Unterricht einzusetzen

Die Entwicklung von Anwendungssoftware für mobile Endgeräte hat in den letzten Jahren massiv an Bedeutung für die Computerbranche gewonnen und Apps werden mehr oder weniger von allen Schülerinnen und Schülern verwendet. Das Smartphone ist zum ständigen Begleiter der Jugendlichen geworden, was nahe legt, die „Einführung in die Programmierung“ so zu gestalten, dass die Ergebnisse der Programmierarbeit sofort auf dem eigenen Gerät bewundert werden können. Die Schülerinnen und Schüler können ihre Arbeit überall mitnehmen und ihren Freunden, Bekannten und ihrer Familie zeigen.

Der App-Inventor ist außerdem intuitiv, einfach zu verwenden, bietet unzählige Möglichkeiten, vermeidet Syntaxfehler, macht Spaß und liefert schnelle Erfolge und tolle Ergebnisse. Es können viele Konzepte der Programmierung vermittelt werden und der spätere Einstieg in eine andere Programmiersprache wird erleichtert. Ein positiver Nebeneffekt ist der Umgang mit der englischen Sprache, dies ist speziell in der Informatik wichtig.

Interessierte finden in der zu diesem Konzept gehörenden Diplomarbeit eine detailliertere Auseinandersetzung mit diesem Thema.

Grundsätzliches zum App Inventor

Der App Inventor wurde vom Massachusetts Institute of Technology (kurz MIT) entwickelt und steht auf www.appinventor.mit.edu kostenlos zur Verfügung. Beim App Inventor handelt es sich um eine Cloud-Anwendung. Die erstellten Projekte werden am MIT-Server gespeichert. Man benötigt lediglich ein Google-Konto. Viele Schüler und Schülerinnen besitzen bereits ein solches. Ansonsten kann auch leicht vorübergehend eines kostenlos erstellt und anschließend wieder gelöscht werden. (siehe Google-Account erstellen und löschen im Anhang)

Mit dem App Inventor hat man die Möglichkeit schnell, relativ einfach und ohne Vorkenntnisse in der Programmierung, mobile Apps für Android zu entwickeln, diese am eigenen Handy zu testen und zu veröffentlichen.

App Inventor ist eine ereignisgesteuerte Programmierungsumgebung. Es handelt sich hierbei um ein visuelles Drag-and-Drop Tool ähnlich wie „Scratch“.

Erst gestaltet man die grafische Oberfläche und programmiert dann das Verhalten der App, indem man „Blöcke“ zusammensetzt. Im Live-Testing kann dann unmittelbar das Ergebnis überprüft werden.

Design -> Programmierung -> Test

Mit dem App Inventor kann man viele technische Features moderner Smartphones in die Programmierung mit einbeziehen. Funktionen wie Sprachausgabe, Kamera, Internetfähigkeit, Sensoren etc. können genutzt werden.

Zielgruppe, Lernziele und Kompetenzen

Zielgruppe

Dieses Konzept ist ausgerichtet auf die Lehr- und Lernziele der 5. Klasse AHS. Es werden

Da es mit dem App Inventor möglich ist, auch sehr komplexe Projekte umzusetzen, ist dieser Leitfaden, mit entsprechenden Erweiterungen, auch für den Einsatz im Wahlfach Informatik geeignet.

Lernziele und Kompetenzen

Die Schülerinnen und Schüler ...

- können Informationsquellen erschließen, Inhalte systematisieren, strukturieren, bewerten, verarbeiten und unterschiedliche Informationsdarstellungen verwenden
- können Konzepte der Informatik verstehen und Methoden und Arbeitsweisen anwenden
- können Algorithmen erklären, entwerfen und darstellen
- können mit dem MIT App Inventor umgehen und Algorithmen implementieren
- können ihre Ergebnisse zur Weiterverarbeitung sichern und präsentieren
- können selbstständig Ideen für Programmierprojekte entwickeln und umsetzen
- können grafische Oberflächen gestalten
- können im Team und in der Gruppe kooperieren
- können Feedback geben und annehmen

Bezug zum Lehrplan Informatik der AHS

Folgende fachliche Kompetenzen des Lehrplanes sollen mit diesem Konzept erfüllt werden:

- „*Informationsquellen erschließen, Inhalte systematisieren, strukturieren, bewerten, verarbeiten und unterschiedliche Informationsdarstellungen verwenden können*“
- „*Begriffe und Konzepte der Informatik verstehen und Methoden und Arbeitsweisen anwenden können*“
- „*Algorithmen erklären, entwerfen, darstellen und in einer Programmiersprache implementieren können*“
- „*Grundprinzipien von Algorithmen, Datenstrukturen und Programmen erklären können*“
(“RIS - Lehrpläne – allgemeinbildende höhere Schulen - Bundesrecht konsolidiert, Fassung vom 18.03.2018,”)

Zu den Fachkompetenzen im Lehrplan für Informatik der 5. Klasse AHS gehört weiters

- die Fähigkeit Kommunikationstechnologien einzusetzen um „[...] *kooperative und kommunikative Arbeitsweisen*“ umzusetzen.

Ebenfalls sollen folgende überfachlichen Kompetenzen erfüllt werden:

- Schülerinnen und Schüler „[...] *sollen ihr kognitives, emotionales und kreatives Potenzial nützen*“

Im Lehrplan steht ebenfalls:

- „die Erweiterung [...] von Sach-, Selbst- und Sozialkompetenz“ ist in den Mittelpunkt zu stellen
- „es ist bei der [...] Gestaltung eines angenehmen und erfolgreichen Lernklimas [...] auf Vertrauen, auf der Förderung der individuellen Stärken und des kreativen Potenzials“ zu achten.
- das Unterrichtsmanagement „hat sich an für Schülerinnen und Schüler transparenten Lehrzielen zu orientieren [...] und soll beispielgebend für die Lern- und Arbeitsorganisation auch außerhalb des Informatikunterrichts sein“
- Schülerinnen und Schülern soll die „[...] Gelegenheit, Neues zu erforschen und bereits Gelerntes in verschiedenen kommunikativen und inhaltlichen Kontexten anzuwenden“, gegeben werden. (ebd.)

Diese Punkte werden im Konzept mithilfe des personenzentrierten Unterrichts umgesetzt.

Auch wird die *Bedeutung gemeinschaftlichen Problemlösens bei der Bearbeitung von Projekten* im Lehrplan thematisiert und bei der Teamarbeit umgesetzt.

Die im Konzept geplanten Gruppengespräche, Präsentationen, das Online-Forum und die Möglichkeit Ergebnisse der Allgemeinheit zur Verfügung zu stellen tragen wie im Lehrplan gewünscht zur Motivation und „Sicherung des Unterrichtsertrages“ bei, da sie unterschiedliche Möglichkeiten haben „[...] ihr Wissen zu präsentieren, sich der Kritik anderer zu stellen und ihre Arbeit zu argumentieren.“ (ebd.)

Den „Leitvorstellungen“ des Allgemeinen Lehrplans für die AHS kann man entnehmen, dass die Schülerinnen und Schüler „[...] zur Übernahme sozialer Verantwortung [...] und „[...] Urteilsbildung [...]“ zu befähigen sind und es dazu erforderlich ist, ein hohes Maß an „[...] Selbstsicherheit sowie selbstbestimmtes, selbst organisiertes Lernen und selbstständiges Handeln zu fördern“ und „[...] zur Teilnahme am sozialen Geschehen anzuhalten“. Unter dem Punkt „Aufgabenbereiche der Schule - Wissensaneignung, Kompetenzerwerb“ ist festgehalten, dass „die Schülerinnen und Schüler im Sinne eines lebensbegleitenden Lernens zur selbstständigen, aktiven Aneignung, aber auch zu einer kritisch-prüfenden Auseinandersetzung mit dem verfügbaren Wissen befähigt und ermutigt werden sollen.“(ebd.). Diese Vorgaben werden mit dem Unterrichtskonzept dahingehend umgesetzt, dass die Schülerinnen und Schüler frei aus dem Informationsangebot wählen können, ihr Arbeitstempo weitgehend selbst festlegen und im Team aber auch als Gruppe zusammenarbeiten. (ebd.)

Im allgemeinen Teil steht: „[...] Schülerinnen und Schüler Fähigkeiten erwerben, die später in Ausbildung und Beruf dringend gebraucht werden, etwa für die Bewältigung kommunikativer und kooperativer Aufgaben“.

die Kommunikationsfähigkeiten der Schülerinnen und Schüler sind dahingehend auszubilden, dass sie in der Lage sind „ihre kognitiven, emotionalen, sozialen und kreativen Kapazitäten zu nutzen und diese auch“ zu erweitern.

Ein weiteres Ziel „des Lernens in der Schule ist [...] die Fähigkeit und Bereitschaft, Verantwortung zu übernehmen, mit anderen“ zusammenzuarbeiten und an der „[...] Gestaltung des sozialen Lebens mitzuwirken“.

Den Schülerinnen und Schülern soll die Möglichkeit gegeben werden, „[...] individuelle Fähigkeiten zu entdecken und [...] sich mit den Ausdrucksformen ihrer Mitmenschen auseinander zu setzen“.

Der kompetenzorientierte „[...] Unterricht ist an den Prinzipien der kommunikativen Didaktik auszurichten [...]“ und „[...] die Förderung sowie die Weiterentwicklung sprachlicher Fähigkeiten [...] ist von besonderer Bedeutung.“ (ebd.)

Voraussetzungen

Voraussetzungen auf Seiten der Lehrperson

Grundkenntnisse des App Inventors und der personenzentrierten Kommunikation.

Alle benötigten Unterlagen sollen vorab auf einer Lernplattform oder einem File-Hosting-Dienst bereitgestellt werden können.

Raumausstattung: PC oder Laptop mit Internetzugang. WLAN wäre zum Testen von Vorteil. Es kann aber auch mit dem Emulator des App Inventors getestet werden. Die dafür nötigen Installationen sollten vorab durchgeführt werden (siehe Vorbereitung und Installation).

Voraussetzungen auf Seiten der Schüler und Schülerinnen

Seitens der Schüler und Schülerinnen bedarf es lediglich Vorkenntnisse in englischer Sprache.

Pro Zweierteam wird ein Google Account benötigt und ein Android Handy oder Tablet zum Testen der Apps wäre sinnvoll. Auch sollten die Schüler und Schülerinnen Kopfhörer mitbringen, um sich beim Ansehen der Videos sowie beim Testen von Tönen in den Anwendungen nicht gegenseitig zu stören.

Ein möglicher Unterrichtsablauf

1. Block

Einführung: Die Lehrperson erklärt den geplanten Ablauf der folgenden Einheiten, um den Schülern und Schülerinnen einen ungefähren Überblick zu verschaffen, was sie erwartet. Es gilt auch zu betonen, dass dieses Modul nicht nur das Programmieren von Apps, sondern auch die Realisierung und Präsentation eines Projekts sowie die Verbesserung der Kommunikationsfähigkeiten zum Ziel hat. Den Schülern und Schülerinnen soll verdeutlicht werden, dass die Lehrperson in den folgenden Einheiten nicht die Funktion des Stoffvermittlers übernimmt, sondern es in ihrer eigenen Verantwortung liegt, die erforderlichen Kompetenzen, aus den ihnen zur Verfügung gestellten Materialien, zu erwerben.

Gruppengespräch: (am besten in einer gesprächsförderlichen Sitzanordnung -> Sesselkreis) Die Themen Programmierung, Apps, die Rolle der Kommunikation in der Informatik, aktives Zuhören etc. können besprochen werden. Dieses Gespräch soll einerseits dazu dienen, sich mit der angemessenen Gesprächskultur vertraut zu machen und andererseits bereits die Möglichkeit bieten, Fachvokabular aus der Informatik unterzubringen und zu besprechen. (ein Arbeitsblatt zum aktiven Zuhören befindet sich im Anhang)

Kennenlernen des App Inventors: Die Schüler und Schülerinnen verschaffen sich mithilfe des Arbeitsblattes_0 und/oder der beiden Einführungsvideos einen ersten Überblick über den App Inventor.

2. Block

Besprechen der Reaktionsblätter: (siehe dazu Lernplattform – Reaktionsblätter) Einige Reaktionen werden von der Lehrperson vorgetragen. Anschließend können Wünsche, Anregungen und Beschwerden der Schüler und Schülerinnen in der Gruppe besprochen werden. Auch hier empfiehlt sich eine gesprächsförderliche Sitzanordnung.

Selbstständiges Arbeiten: Die Schüler und Schülerinnen erstellen mithilfe der ihnen zur Verfügung gestellten Unterlagen und Informationsquellen erste kleine Programme und dokumentieren in Stichworten ihre Vorgehensweise. Dies kann sowohl als Einzelarbeit als auch im Team erfolgen. Die Lehrperson steht während dieser Phase als Berater zur Verfügung.

Kurzpräsentationen mit Feedback: Jedes Team präsentiert in wenigen Worten, was in dieser Einheit umgesetzt werden konnte und erhalten ein kurzes Feedback von den Kollegen und Kolleginnen sowie von der Lehrperson.

3. Block

Besprechen der Reaktionsblätter: (siehe Block 2). Im Anschluss können erste Erfahrungen und eventuelle Probleme aufgegriffen werden.

Einsatzbesprechung („Daily Standup-Meeting“): Jedes Team gibt das Tagesziel kurz bekannt.

Selbstständiges Arbeiten: Weitere Arbeitsblätter und Informationsquellen werden von den Schülern und Schülerinnen bearbeitet. Dabei bleibt es den Schülern und Schülerinnen überlassen, ob sie die Zusatzaufgaben selbstständig oder mit der beigefügten Unterstützung lösen. Ein selbstständiges Erarbeiten der Aufgaben sollte jedoch angeregt werden.

Kurzpräsentationen mit Feedback: (siehe Block 2)

Anmerkung: Dieser Block sollte bei Bedarf wiederholt werden.

4. Block

Besprechen der Reaktionsblätter: (siehe Block 2)

Teambesprechung: Jedes Team skizziert eine Projektidee, welche mit dem App Inventor in den nächsten Einheiten umgesetzt werden soll, oder plant die Erweiterung einer Anwendung aus den Arbeitsblättern. Dieser Vorgang bedingt eine Reflexion der eigenen Fähigkeiten und Interessen. Er bietet aber auch die Möglichkeit die Schwierigkeit des Projektes je nach Leistungsstand und Vorwissen selbst festzulegen. Durch die freie Wahl des Projektthemas kann eine größere Identifikation mit der Arbeit erzielt werden und somit die Motivation gesteigert werden.

Präsentation der Projektideen mit Feedback („Elevator Pitch“): Die Teams geben eine **kurze** Zusammenfassung ihrer Idee und „verkaufen“ das Projekt. Das anschließende Feedback soll den Teams weitere Anregungen liefern, auf mögliche Schwierigkeiten aufmerksam machen und helfen weitere Entscheidungen zu treffen. Hier hat die Lehrperson die Möglichkeit zu komplexe Ideen vorerst zu reduzieren bzw. Anregungen für den Ausbau zu einfacher Projekte zu liefern.

Selbstständiges Arbeiten: Die Teams verfeinern mit den Informationen aus dem Feedback ihre Projektidee und beginnen mit der Implementierung.

Kurzpräsentationen mit Feedback: (siehe Block 2)

5. Block

Besprechen der Reaktionsblätter: (siehe Block 2)

Einsatzbesprechung („Daily Standup-Meeting“): Jedes Team gibt das Tagesziel kurz bekannt und hat die Möglichkeit Schwierigkeiten mit der Gruppe zu besprechen.

Selbstständiges Arbeiten: Die Teams arbeiten weiter an ihren Projekten. Durch ihr Ziel, ein gutes Endprodukt zu erhalten, wird die Selbstkoordination der Schüler und Schülerinnen gefördert. Eine projektorientierte Arbeitsform wird durch die Aufforderung zur Dokumentation, durch die Einsatzbesprechungen und die Teamarbeit gefördert.

Kurzpräsentationen mit Feedback: (siehe Block 2)

Anmerkung: Dieser Block sollte bei Bedarf beliebig oft wiederholt werden.

6. Block

Besprechen der Reaktionsblätter: (siehe Block 2)

Präsentation der abgeschlossenen Projekte mit Feedback: Die fertigen Apps werden von den Schülern und Schülerinnen präsentiert.

Gruppengespräch: Zur weiteren Ergebnissicherung wird über die gesammelten Erfahrungen und die Vor- und Nachteile des App Inventors gesprochen. Dies sollte unter anderem eine Motivation liefern, dass sich die Schüler und Schülerinnen auch weiterhin mit Programmierung auseinandersetzen und vielleicht sogar andere Programmiersprachen kennenlernen wollen.

Ergebnissicherung:

Die Ergebnissicherung erfolgt einerseits über die Teampräsentationen und über die Gruppengespräche und andererseits über die Apps, welche die Schüler und Schülerinnen auf die gemeinsam genutzte Lernplattform (als .aia_Datei) laden sollen.

Einstieg in den App Inventor 2

Systemanforderungen:

Hinweis: **Der Internet Explorer wird nicht unterstützt.** Es wird Chrome oder Firefox empfohlen.

Computer und Betriebssystem

Windows:	Windows XP, Windows Vista, Windows 7 oder höher
Macintosh (mit Intel-Prozessor):	Mac OS X 10.5 oder höher
GNU / Linux:	Ubuntu 8 oder höher, Debian 5 oder höher

Anmerkung: GNU / Linux-Live-Entwicklung unterstützt zum Testen nur WiFi-Verbindungen zwischen Computern und Android-Gerät

Smartphone oder Tablet (es kann aber auch der Emulator zum Testen benutzt werden)
Android Betriebssystem 2.3 („Gingerbread“) oder höher

(„Setting Up App Inventor | Explore MIT App Inventor“)

Vorbereitung und Installation:

Der Designer und der Block-Editor laufen komplett im Browser. („Einrichten App Inventor | Entdecken MIT App Inventor“)

Aufruf der Webseite <http://appinventor.mit.edu/explore/> und „**Create apps!**“ auswählen.



Abbildung 1 – Startseite appinventor.mit.edu/explore

Nachdem man sich mit einer Google-Mail-Adresse angemeldet hat, wird man zur Startseite des App-Inventors weitergeleitet.

Alternativ kann man auch direkt mit <http://ai2.appinventor.mit.edu/> zum App-Inventor gelangen. Nur für das Live-Testing sind Installationen erforderlich. Hierfür hat man drei Möglichkeiten.

Möglichkeit 1 (bevorzugte Möglichkeit)



Abbildung 2 - Testen am Handy mit WLAN

Es wird nur ein Android-Handy oder Tablet und eine drahtlose Internetverbindung benötigt. Es muss keine Software auf dem Computer installiert werden.

Lediglich die Installation der App Inventor Companion App auf dem Android Handy ist erforderlich. (Download vom Google Play Store)



Abbildung 3 -Companion App

Sobald die App installiert ist, können Projekte aus dem App Inventor mit dem Handy live getestet werden. Handy und Computer müssen sich dazu im gleichen WLAN befinden.

Im App Inventor öffnet man ein Projekt und wählt dann „Connect“ und „AI Companion“ aus dem Hauptmenü aus:

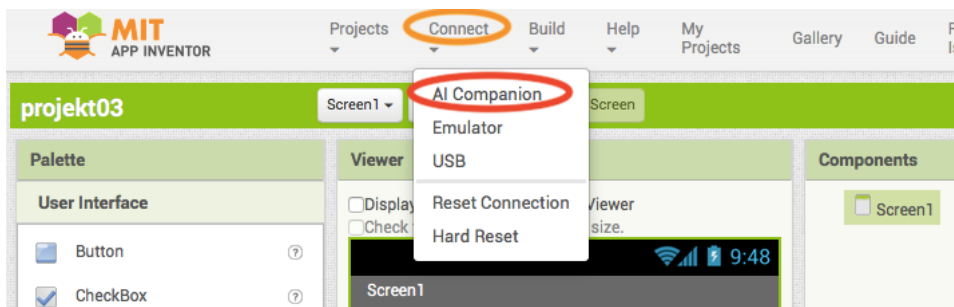


Abbildung 4 – Verbindungsfenster

Ein Dialog mit einem QR-Code wird auf dem PC-Bildschirm angezeigt.

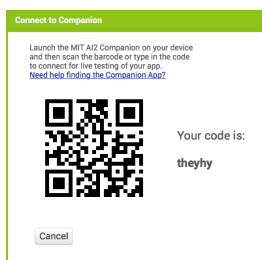


Abbildung 5 - QR-Verbindungscode

Auf dem Android Gerät startet man nun die MIT Companion App und klickt auf die „**Scan QR Code**“ – Taste auf der Companion-App und scannt den Code aus dem App Inventor-Fenster.

Innerhalb von wenigen Sekunden sollte man die App sehen, welche auf dem Computer entwickelt wurde. Eine Aktualisierung erfolgt immer wenn Änderungen am Design oder den Blöcken gemacht wird. Dieses Verhalten wird „Live-Testing“ genannt.

Möglichkeit 2



Abbildung 6 - Testen am Computer

Sollten keine Androidgeräte zur Verfügung stehen, ist diese Möglichkeit sinnvoll.

Verwendung des On-Screen Android – Emulators.

Dazu muss einige Software auf dem Computer installiert werden.

Genaue Anweisungen hierzu sind auf: <http://appinventor.mit.edu/explore/ai2/windows.html> zu finden.

Sollte diese Möglichkeit bevorzugt werden, ist es ratsam den Emulator vor der ersten Einheit auf den Computern zu installieren und diesen während der Sitzungen nicht zu schließen, da der Start des Emulators einige Minuten in Anspruch nehmen kann.

Möglichkeit 3



Abbildung 7 - Testen am Handy mit USB-Verbindung

Auch bei dieser Möglichkeit ist es notwendig Software auf dem Computer zu installieren. Genaue Informationen findet man ebenfalls unter: <http://appinventor.mit.edu/explore/ai2/windows.html>

Apps verpacken und teilen:

Apps können sowohl in ausführbarem Format (.apk) als auch in Quellcodeformat (.aia) verpackt werden bzw. am Google Play Store bereitgestellt werden.

Für den Schulunterricht bietet sich das Quellcodeformat an, da dies leicht in den App Inventor geladen und von der Lehrperson überprüft werden kann. Es kann auch für Kollegen und Kolleginnen zum Verändern bereitgestellt werden.

Die exportierte Datei befindet sich dann in den Downloads ihres Computers und kann per E-Mail verschickt oder auf einer Lernplattform Verfügung gestellt werden.

Mit **Import project (.aia) from my computer** kann eine bereitgestellte „.aia-Datei“ in den App Inventor geladen und bearbeitet werden.

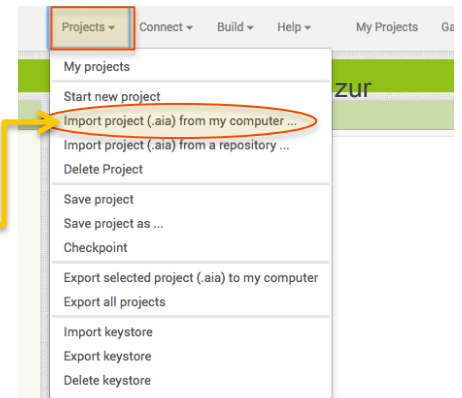


Abbildung 8 – Projektexport und -import

Der App Inventor im Überblick

Das Start- bzw. Designfenster (siehe Abbildung 8) ist in vier Bereiche unterteilt:

Im Designfenster werden die gewünschten sichtbaren, aber auch unsichtbaren, funktionalen Elemente der App, auf einem virtuellen Smartphone-Bildschirm platziert.

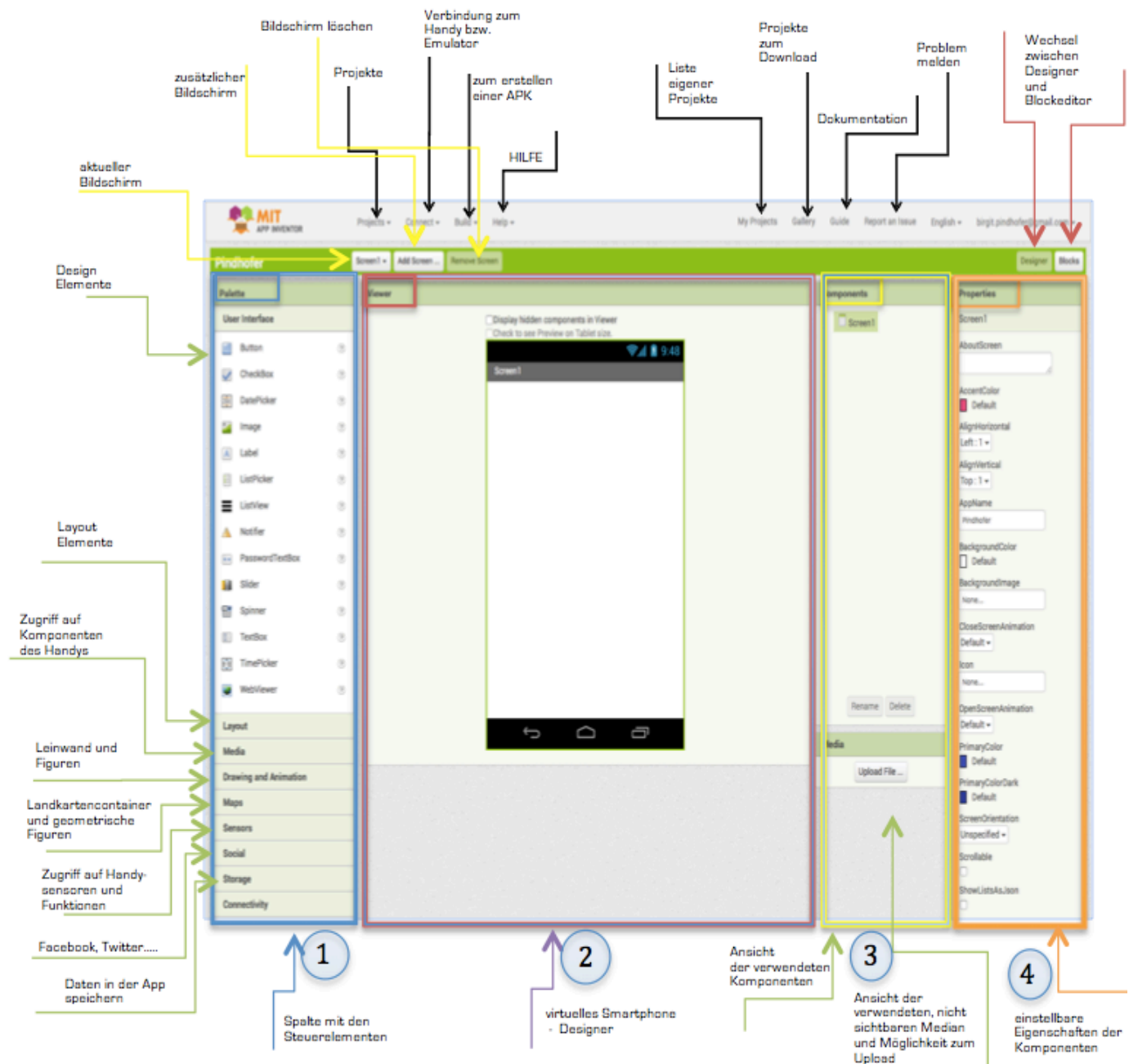


Abbildung 9 – Start- bzw. Designfenster

1. Palette:

Steuerelemente zum Erstellen der App. Dieser Bereich ist wiederum in Abschnitte unterteilt:

- **User Interface:** grundlegende Schaltflächen wie Buttons und Textfelder. Dieser Bereich ist bereits geöffnet, wenn man den Designer startet.
- **Layout:** Komponenten zum Anordnen der Elemente am Bildschirm (horizontal, vertikal oder in Tabellenform)
- **Medien:** Komponenten zum Aufnehmen und Abspielen von Fotos, Videos, Tönen, zur Spracherkennung und zur Umwandlung von Text in Sprache
- **Drawing and Animation:** Elemente zum Erstellen einfacher Zeichnungen und Animationen
- **Sensors:** Komponenten die den Zugriff auf den Beschleunigungs-, Ortungs-, NFC-Sensor ermöglichen, sowie einen Barcode-Scanner
- **Social:** Elemente die mit der Kontaktliste des Telefons zum Senden von Nachrichten, Tätigen von Anrufen und der Kommunikation mit sozialen Netzwerken arbeiten
- **Storage:** Komponenten zum Speichern von Daten lokal auf dem Handy oder auf einem Webserver
- **Connectivity:** Elemente zum Starten externer Anwendungen, zur Verbindung mit Bluetooth-Geräten, für einen Internetzugang und um Lego-Mindstorm ® Roboter anschließen zu können.

2. Viewer:

Hier wird der sichtbare Teil der App gestaltet, indem man die Elemente aus der Palette auf das virtuelle Smartphone-Display zieht und nach Belieben anordnet. (**Achtung:** das Layout kann auf dem Endgerät

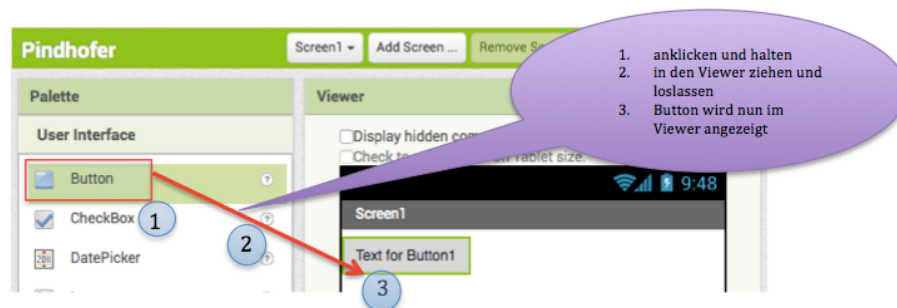


Abbildung 10 - Komponenten zum Viewer hinzufügen

etwas anders aussehen. Daher immer live testen)

3. Components:

Anzeige bereits verwendeter Steuerelemente, in hierarchischer Struktur. Klickt man ein Element im Viewer an, so wird diese Komponente farbig unterlegt. In diesem Bereich kann man Elementen einen neuen Namen geben (ratsam um den Überblick leichter zu behalten) oder sie löschen.

Media: befindet sich direkt unter Components und dient zum Hochladen und Verwalten von Bild-, Video- und Audiodateien.

4. **Properties**

Einstellungen, welche die Eigenschaften der Elemente festlegen. Diese Eigenschaften können in diesem Bereich individualisiert werden.

Durch einen Klick auf „**My Projects**“ legt man ein neues Projekt über die Schaltfläche „**New Project**“ an und kann unmittelbar mit der Erstellung der eigenen App anfangen.

Rechts oben am Bildschirm befinden sich zwei Schaltflächen „**Designer**“ und „**Blocks**“ zum Wechseln zwischen der Designer-Ansicht und dem Blocks-Editor.

Im Blocks-Editor (siehe Abbildung 9) wird das Verhalten der App programmiert.

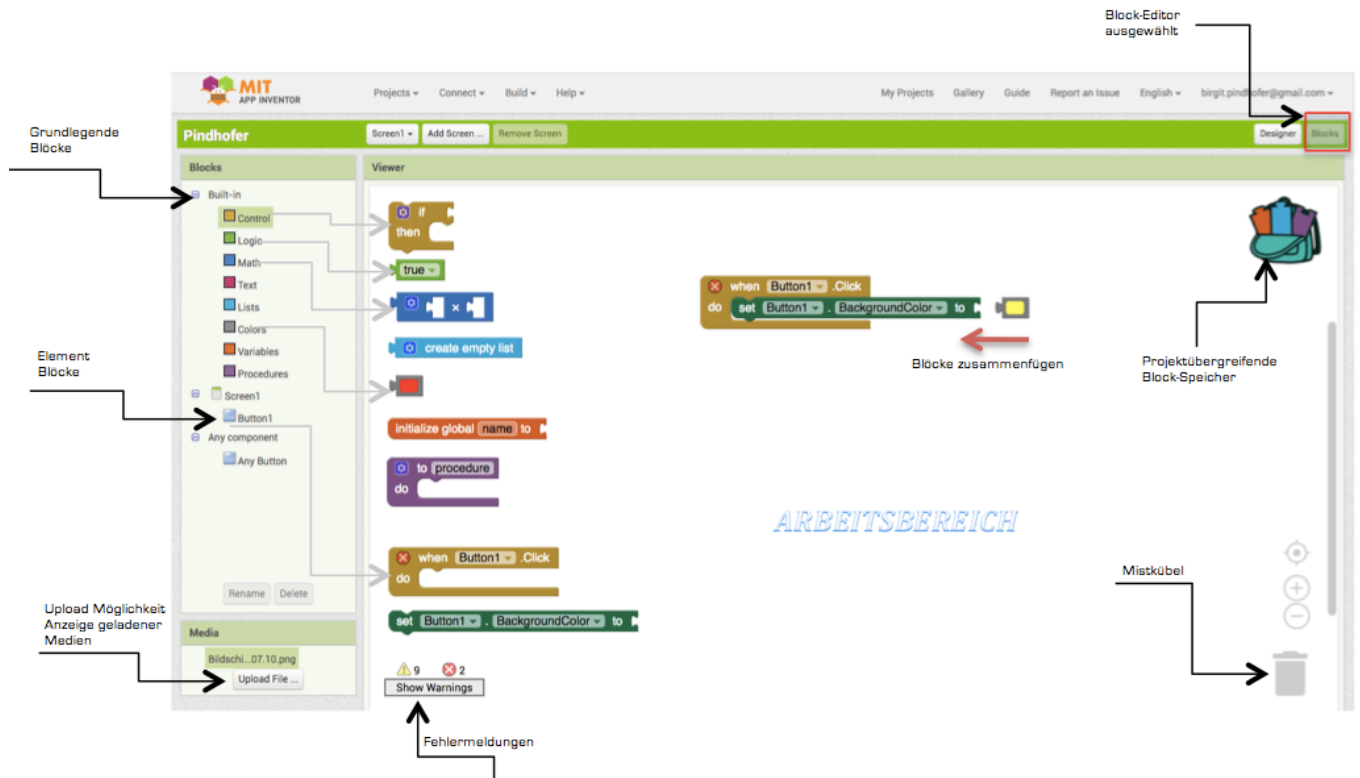


Abbildung 11 - Blocks Editor

Die Spalte auf der linken Seite des Blocks-Editors bietet Zugriff auf die Blöcke, die man verwenden kann und ist in mehrere Abschnitte unterteilt. Jeder von diesen stellt ein getrenntes Set an Bausteinen zur Verfügung.

- **Built-In:** Die erste Kategorie enthält all jene Blöcke zum definieren der Programmlogik. Sie ist in folgende Unterkategorien gegliedert:
 - Control:** Blöcke, die den Programmablauf steuern (do, if, for each, while ...)
 - Logic:** Blöcke wie true, false, =, and, or
 - Math:** Blöcke, das Verarbeiten von Zahlen und das Rechnen mit Zahlen ermögliche
 - Text:** Blöcke die Texte verarbeiten
 - Lists:** Blöcke zum Erstellen und Bearbeiten von Listen
 - Colors:** Blöcke zum Verwenden und Mischen von Farben
 - Variables:** Blöcke zum Initialisieren und Verarbeiten von Variablen
 - Procedures:** Blöcke zum Erstellen eigener Prozeduren



- **Screen1:** Wenn im Designer eine Komponente in den Screen1 gezogen wurde, wird diesem Abschnitt ein Satz von Komponentenblöcken hinzugefügt. Komponentenblöcke sind Blöcke, die eine Aktion für ein bestimmtes Element ausführen oder Zugriff auf dessen Eigenschaften bieten.
- **Any component:** Dieser Abschnitt enthält erweiterte Blöcke, die es ermöglichen mit einer beliebigen Komponente in der App zu arbeiten.

Durch Anklicken einer Unterkategorie erhält man eine Liste von Blöcken. Mittels Rechtsklick auf einen Puzzlestein gelangt man direkt zum passenden Hilfedokument. Blöcke haben die Form von Puzzleteilen und können zu einem Programm zusammengefügt werden.

Jeder Block beschreibt eine Aktion, welche ausgeführt wird, sobald der betreffende Block im Ablauf erreicht wird. Nachdem diese ausgeführt wurde, folgt die des nächsten Blocks. Man kann die Blöcke leicht mit der Maus in den Arbeitsbereich ziehen.

Hier hat man, mittels Rechtsklick auf einen Block, die Möglichkeit diesen zu vervielfältigen, zu löschen, zu deaktivieren bzw. wieder zu aktivieren.

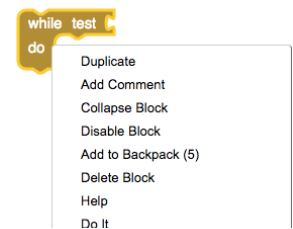



Abbildung 12 - Blockmenü

Blöcke, welche nicht mehr gebraucht werden, kann man in den Mülleimer verschieben bzw. mit der Taste „Löschen“ entfernen.

Fehlermeldungen und Warnungen werden im unteren Bereich des Viewers angezeigt. Auf den betroffenen Blöcken ist ein  zu sehen. Klickt man darauf, so erhält man eine Beschreibung des Fehlers.

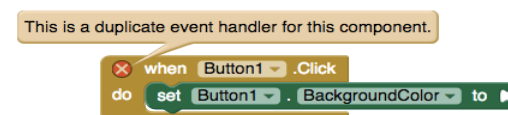


Abbildung 13 - Fehlermeldung

Für die Schüler und Schülerinnen sind die wichtigsten Informationen auf dem Arbeitsblatt_0 zusammengefasst. Zusätzlich sollte der Link zum Video ([App_inventor Einstieg I](#)) und [App_Inventor Einstieg II](#) bereitgestellt werden. Ebenso können sich die Schüler und Schülerinnen das Tutorial auf der App Inventor Seite ansehen.

Arbeitsblatt 0

DER APP INVENTOR

Einstieg

Öffne bitte mit diesem Link den App Inventor: <http://appinventor.mit.edu/explore/> oder tippe diese Adresse selbst ein und wähle „Create app!“ aus

MIT APP INVENTOR About News & Events Resources **Create app!**

Anyone Can Build Apps That Impact the World

Google Custom Search

Donat!

Logge dich mit deinem Google Account ein!
Verwende dazu deine Gmail-Adresse und das dazugehörige Passwort.

Google

Anmeldung

Mit dem Google-Konto anmelden.

E-Mail-Adresse eingeben

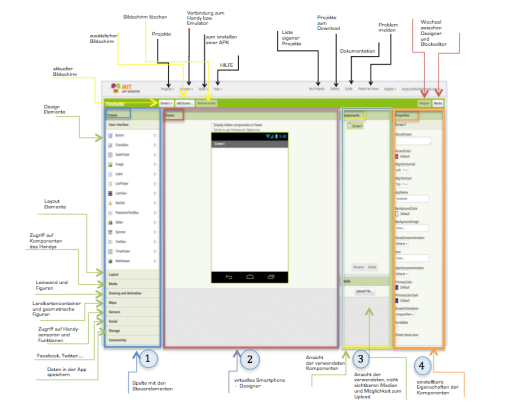
[E-Mail-Adresse vergessen?](#)

Konto erstellen

DER APP INVENTOR

Hier ein kleiner Überblick über den Designeditor.

Im Designeditor gestaltest du das Aussehen deiner App und fügst alle Komponenten hinzu die du brauchst. Auch solche die man später nicht sieht, so wie zu Beispiel Töne



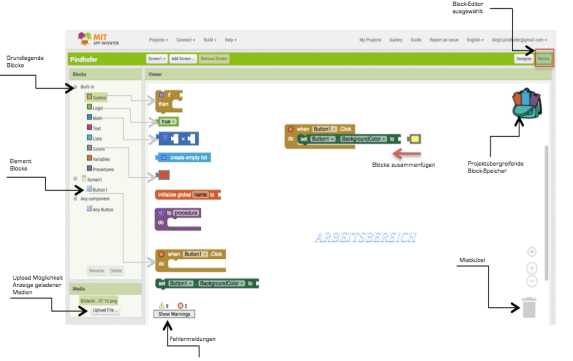
So wechselst du zwischen Design- und Blockeditor
Klicke auf Blocks wenn du in den Blockeditor wechseln willst bzw. auf Designer wenn du wieder in den Designeditor zurück willst

Hier kannst du zwischen Design- und Blockeditor wechseln

DER APP INVENTOR

Hier ein kleiner Überblick über den Blockeditor.

Im Blockeditor programmiert du das Verhalten deiner App



Block-Editor ausgewählt

Grundlegende Blöcke

Element-Blöcke

Upload Möglichkeit: Anzeige geladener Medien

Informations

Block-Editor ausgewählt

Blöcke zusammenfügen

Projektübergreifende Block-Speicher

ARBEITSBEREICH

DER APP INVENTOR

So erstellst du ein neues Projekt

Projects Connect Build Help

My projects

Start new project

Import project (.aiia) from my computer ...

Import project (.aiia) from a repository ...

Delete Project

Create new App Inventor project

Project name

Cancel OK

Gib deinem Projekt einen Namen und bestätige diesen mit OK

So kannst du Komponenten zu deiner App hinzufügen

Pindhofer

Palette

User Interface

Button

Checkbox

DatePicker

1. anklickes und haltes in den Viewer ziehen und loslassen

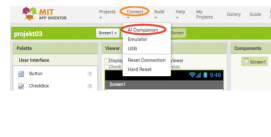
2. Button wird nun im Viewer angezeigt

3. Text für Button!

So kannst du deine App auf deinem Android-Handy testen

Du musst die App Inventor Companion App aus dem Android App Store auf deinem Smartphone installieren und diese dann starten. Dann brauchst du nur noch auf „Scan QR-Code“ und den Code deiner App einzuscannen.

Wie du zum Code kommst, siehst du hier:



Ein Dialog mit einem QR-Code wird auf dem PC-Bildschirm angezeigt.

Die Lernplattform

Auf der Lernplattform werden die für das Selbststudium benötigten Unterlagen bereitgestellt. Hierbei handelt es sich um Arbeitsblätter, Videos und Links zu Websites, welche den Schülern und Schülerinnen helfen sollen, das benötigte Wissen selbstständig und in eigener Geschwindigkeit zu erwerben.

Die Schüler und Schülerin geben auch ihre Projekte und Reaktionsblätter, für alle einsehbar, auf der Lernplattform ab.

Arbeitsblätter

Die Arbeitsblätter, welche im Anhang zu finden sind, beinhalten folgende Elemente:

1. die Gestaltung der jeweiligen App im Designer wird vorgenommen.
2. Programmierungen des Verhaltens der Elemente im Block-Editor
3. Tipps und Theorieinput
4. Aufgaben zur Vertiefung

Videos und Internetlinks

Eine Sammlung von Videos und Internetlinks befindet sich im Anhang.

Da die Erstellung aller benötigten Videos zu den Arbeitsblättern den Rahmen dieser Arbeit sprengen würde, befinden sich einige Tipps zur Herstellung von Erklärvideos ebenfalls im Anhang.

Reaktionsblätter

Jede Einheit soll von den Schüler und Schülerinnen schriftlich reflektiert werden. Hierbei sollte es sich jedoch nicht um eine Kurzzusammenfassung der letzten Einheit handeln, sondern die Schüler und Schülerinnen sollten sowohl positive als auch negative Eindrücke beschreiben. Dieses Reaktionsblatt soll dann, bis ein paar Tage vor der nächsten Zusammenkunft, auf der gemeinsam genutzten Lernplattform der Allgemeinheit zur Verfügung gestellt werden. Dieses Feedback gibt der Lehrperson die Möglichkeit das Konzept auf die Bedürfnisse der Schüler und Schülerinnen anzupassen, sowie auf etwaige Probleme und Wünsche einzugehen.

Zu Beginn der folgenden Einheit werden die Reaktionen besprochen, was nicht nur einen guten Einstieg bietet, sondern den Schülern und Schülerinnen auch eine Atmosphäre der Freiheit, des Verstehens und der Akzeptanz vermittelt. (Motschnig-Pitrik, Kabicher, & Figl, o. J.)

Die Arbeitsblätter und die darin verwendeten Medien stehen zum Download zur Verfügung.

[Unterlagen zum Download](#)

Anhang

Arbeitsblatt aktives Zuhören

WIE HÖRE ICH AKTIV ZU?



Ich zeige Interesse!!!

Du hast meine ungeteilte Aufmerksamkeit!

Ich bin neugierig was du zu sagen hast!

Ich bin dir zugewandt und sehe dir in die Augen!

Ich unterbreche dich nicht!

Ich nicke mit dem Kopf!

Ich achte auf meinen Gesichtsausdruck!

„Aah..“

„Mhm..“

„Ich verstehe..“

„interessant..“

Ich gehe sicher, dass ich dich verstanden habe!!!

Ich stelle dir Verständnisfragen um Unklarheiten zu vermeiden!

Ich wiederhole das Gesagte mit eigenen Worten!

Ich fasse kurz zusammen, damit du noch was ergänzen kannst!



„Habe ich das richtig verstanden, dass..“

„Meinst du damit...“



Ich spreche deine Emotionen an!!

„Ich kann sehen, dass dich das ärgert...“

„Das hat dir Spass gemacht...“



Bildquelle:

<https://pixabay.com/de/besprechung-meeting-gespr%C3%A4ch-1002800/> <https://pixabay.com/de/ohr-ohrmuschel-hören-zuhören-2973126/> <https://pixabay.com/de/emoji-emotion-emoticon-ausdruck-1585197/> <https://pixabay.com/de/zitat-sprechblase-anf%C3%BChrungszeichen-1122421/> <https://pixabay.com/de/fragezeichen-alphabet-frage-3246711/> <https://pixabay.com/de/smiley-freude-umarmen-2674401/>

Arbeitsblatt 1 und Arbeitsblatt 2

Diese Beispiele dienen vor allem dazu, den App Inventor kennenzulernen. Es werden erstmals Buttons verwendet und Medien (Fotos und Töne) hochgeladen um diese in der App verwenden zu können.

Die Schüler und Schülerinnen erstellen ein neues Projekt, gestalten das Aussehen ihrer Anwendung und programmieren erste Funktionen mit einem „**when-Befehl**“.

In den weiteren Schritten wird geübt, dass man das Aussehen von Buttons und Canvas verändern kann. Es werden auch bereits „**if-then-else-Anweisung**“ verwendet.

Lernziele:

Schüler und Schülerinnen

- lernen den App-Inventor kennen
- erstellen eine erste eigene App, indem sie Komponenten auswählen und ihnen dann sagen, was zu tun und wann es zu tun ist.
- Verwenden den Komponenten-Designers
- ändern das Aussehen der verwendeten Komponenten
- Fügen Medien (Sounds und Bilder) zur App hinzu
- Arbeiten im Blockeditor zum Zusammenstellen von Blöcken
- verwenden Schleifen
- Testen der App mit Live-Tests von App Inventor

A R B E I T S B L A T T 1



1.1 Happy-Sad APP

Öffne bitte den App Inventor: <http://appinventor.mit.edu/explore/>
und starte ein neues Projekt: Projects -> Start new project

Tipp: Du brauchst dich nicht um die Speicherung deines Projektes zu kümmern – Projekte werden automatisch gespeichert

Ziel

Eine App die ein lachendes oder ein weinendes Gesicht zeigt.
Welches Gesicht angezeigt wird hängt davon ab, welcher Button gedrückt wurde.

Rechts siehst du das Design der App

Welche Komponenten brauchst du dazu?

User Interface

1 x Image
2 x Button

Bild- und Sounddateien für diese App findest du auf
https://drive.google.com/Bilder_Töne

Tipp: Ändere die Namen deiner Buttons!! Wähle dazu Bezeichnungen, die dir helfen zu wissen wofür du sie verwenden willst. Speziell bei größeren Projekten ist das wichtig. Es hilft dir den Überblick zu behalten. (im Beispiel wurde „BSad“ für den Button „sad“ und „BSmile“ für den Button „smile“ gewählt.)

Tipp: Um Bilder und Sounds verwenden zu können, musst du diese zuerst hochladen

Wie kannst du das Ziel erreichen?

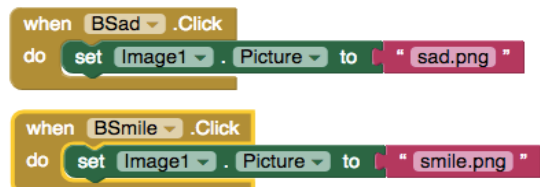
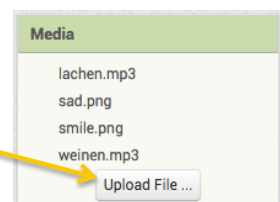
Ziehe die benötigten Komponenten im Designfenster auf den virtuellen Bildschirm, indem du sie anklickst und mit gedrückter Maustaste auf den Bildschirm ziehst und dort wieder loslässt.
Die Eigenschaften deiner Komponenten kannst du in die „**Properties**“ nach Belieben ändern. (z.B. Schriftgröße, Text des Buttons, Farbe etc.)

Im Blockeditor kannst du nun den Programmcode zusammenbauen.

Vergiss bitte nicht, deine App zu testen!



Durch Klicken von glücklich auf traurig umschalten



Aufgabe: Beim Klicken eines Buttons soll nun nicht nur das Bild geändert werden, sondern es soll auch ein passender Ton abgespielt werden. Die Sounddateien dazu musst du noch hochladen, falls du das nicht schon gemacht hast.

Versuche es zuerst selbst. Falls du Schwierigkeiten hast, findest du auf der nächsten Seite eine Lösung.

Tipp: Den passenden Befehl findest du unter **Blocks -> Screen1 -> SSmile bzw SSad**

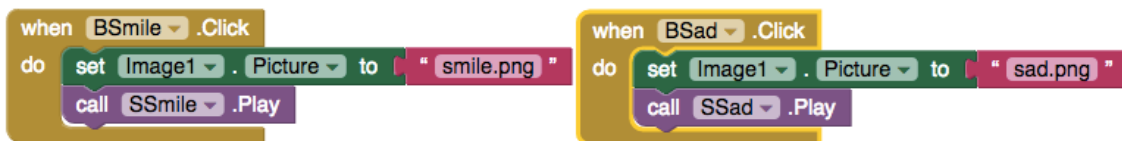
1.2 Happy-Sad APP mit Sound

Welche Komponenten brauchst du zusätzlich?

Media

2 x Sound

Bild- und Sounddateien für diese App findest du auf https://drive.google.com/Bilder_Töne



Vergiss bitte nicht, deine App zu testen!

Du hast schon eine Menge gelernt!

- Du hast eine App erstellt, indem du Komponenten ausgewählt und ihnen gesagt hast was sie wann und wo zu tun haben
- Du hast den Designer zum Auswählen von Komponenten verwendet und festgestellt, dass einige Komponenten auf dem Telefonbildschirm zu sehen sind und andere nicht.
- Du hast Medien (Sounds und Bilder) zu deiner App hinzugefügt, indem du sie hochgeladen hast
- Du hast im Blockeditor das Verhalten deiner Komponenten definiert.
- Du hast deine Apps mit Live-Tests von App Inventor getestet.

BRAVO!!!

ARBEITSBLATT 2

2.1 Happy-Sad APP ohne Buttons

Öffne bitte den App Inventor: <http://appinventor.mit.edu/explore/>
und starte ein neues Projekt: Projects -> Start new project

Ziel

Eine App die ein lachendes oder ein weinendes Gesicht zeigt.
Der Wechsel zwischen den Gesichtern wird hier aber nicht durch Buttons gesteuert, sondern durch einen Klick auf das Gesicht.

Rechts siehst du das Design der App

Welche Komponenten brauchst du dazu?

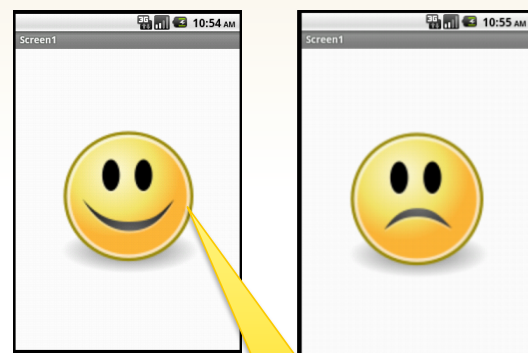
User Interface

1 x Button

Media

2 x Bild

Bilddateien für diese App findest du auf
https://drive.google.com/Bilder_Töne

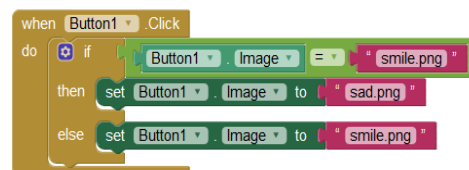


Durch Klicken von glücklich auf traurig umschalten

Wie kannst du das Ziel erreichen?

Das Aussehen des Buttons wird in den Properties so geändert, dass er ein „Gesicht“ ist. Du findest auch hier die Möglichkeit ein Bild zu verwenden.

Im Blocks-Editor gibst du nun an, dass wenn der Button gedrückt wird, und das Bild „smile.png“ zu sehen ist, dann soll das Bild auf „sad.png“ geändert werden. Ansonsten (das heißt, wenn gerade das Bild „sad.png“ zu sehen ist) soll es auf „smile.png“ geändert werden



Vergiss bitte nicht, deine App zu testen!

Aufgabe: Nun soll der Wechsel zwischen den Gesichtern durch „wischen“ realisiert werden. Wenn du über das Gesicht nach unten wischst soll es traurig werden. Wenn du auf dem Gesicht nach oben wischst soll es glücklich werden.

Tipp: Verwende statt dem Button eine „Leinwand“ (Canvas)

„when Canvas1. Dragged“

Versuche diese Aufgabe selbst zu lösen bevor du dir eine mögliche Lösung unter 2.2. ansiehst.

2.2 Happy-Sad APP mit “Wischen”

Welche Komponenten brauchst du?

User Interface

1 x Canvas

Media

2 x Bild

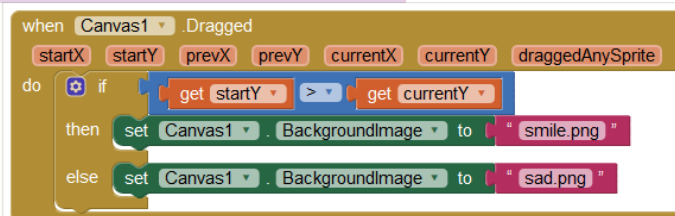
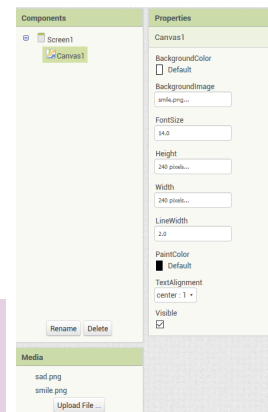
Bilddateien für diese App findest du auf https://drive.google.com/Bilder_Töne

Wie kannst du das Ziel erreichen?

Das Aussehen des Canvas wird in den Properties so geändert, dass er ein „Gesicht“ ist. Du findest auch hier die Möglichkeit ein Bild zu verwenden.

Im Blocks-Editor gibst du nun an, dass wenn über die Canvas von oben nach unten gewischt wird, und das Bild „smile.png“ zu sehen ist, dann soll das Bild auf „sad.png“ gesetzt werden. Ansonsten (das heißt, wenn von unten nach oben gewischt wird) soll es auf „smile.png“ geändert werden.

Tipp: die Koordinaten im App Inventor verhalten sich nicht wie gewohnt, (0/0) befindet sich beim App Inventor in der linken oberen Ecke (d.h. wenn der Startpunkt beim Wischen größer ist als der Endpunkt -> hast du nach oben gewischt)



Vergiss bitte nicht, deine App zu testen!

Du hast schon wieder eine Menge gelernt!

- Du hast das Aussehen deiner ausgewählten Komponenten verändert
- Du hast mit Bedingungen gearbeitet und den Befehl if-then-else verwendet
- Du hast mit dem Koordinatensystem des App Inventors gearbeitet

BRAVO!!!

Arbeitsblatt 3

In diesem Beispiel wird der Umgang mit dem App Inventor vertieft, sowie die Möglichkeiten für verschiedene Anordnungen der Komponenten am Bildschirm erprobt.

Außerdem werden Variablen thematisiert und eine Datei verwendet, um eine Zeichnung auf dem eigenen Handy zu speichern.

In der Erweiterung wird ein Popup-Fenster verwendet, um Benutzereingaben für die Speicherung zu verarbeiten.

Lernziele:

Schüler und Schülerinnen

- vertiefen den Umgang mit dem App-Inventor
- lernen weitere Komponenten kennen
- arbeiten mit den Properties um das Aussehen der Komponenten anzupassen
- verwenden Variablen und Dateien
- speichern Daten

Die Idee zur App stammt aus dem Buch Android Apps mit Appinventor2: Jeder kann programmieren, Karl-Hermann Rollke.

ARBEITSBLATT 3



Malen

Öffne bitte den App Inventor: <http://appinventor.mit.edu/explore/>
und starte ein neues Projekt: Projects -> Start new project

Ziel

Eine App mit der du in den Farben Rot, Grün, Blau und Schwarz malen kannst. Die Farbwahl geschieht über Buttons. Dein Bild kann in einer Datei gespeichert werden („Button Speichern“) und mit „Button Laden“ wieder geladen werden. „Button Löschen“ löscht nur den Teil, der nach dem Laden gemalt wurde. „Button Alles Löschen“ löscht das ganze Bild aber nicht die Datei.

Rechts siehst du das Design der App

Welche Komponenten brauchst du dazu?

User Interface

8 x Button
1 x Canvas

Layout

2 x HorizontalArrangement

Storage

1 x File

Durch Klicken auf die Buttons wählst du die Farbe aus – du zeichnest mit dem Finger am Display des Handys



Nicht sichtbare Komponente um Dateien auf deinem Gerät zu speichern oder zu lesen. So bleibt der Inhalt erhalten, auch wenn du die App schließt

Wie kannst du das Ziel erreichen?

Das Aussehen des Buttons wird in den **Properties** so geändert, dass sie der Farbe entsprechen, welche sie repräsentieren. Zum Anordnen der Buttons am Bildschirm verwende „Layout -> HorizontalArrangement“ zwei mal um zwei Reihen mit Buttons realisieren zu können.

Um auf dem „Canvas“ malen zu können, wähle die gewünschte Farbe über die Buttons, „wische“ über den Bildschirm und rufe für den „Canvas“ die fertige **Funktion** „DrawLine“ auf



Versuche nun selbst diese Aufgabe umzusetzen.
Auf der nächsten Seite siehst du dann eine mögliche Lösung!

Tipp: Eine Variable repräsentiert einen im Computer gespeicherten Wert.
Der App Inventor hat zwei Arten von Variablen.

Globale Variable: alle Blöcke können auf diese Variable zugreifen.

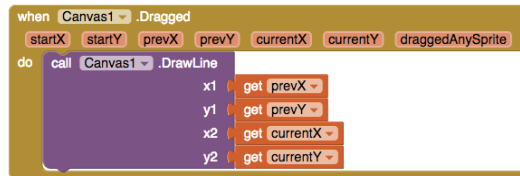
Lokale Variable: gelten nur innerhalb einer Methode oder Funktion. Es können nur Blöcke darauf zugreifen, welche zu dazu gehören.

Neue Variablen müssen initialisiert werden. Das heißt, dass ihnen ein Startwert zugewiesen werden muss.

Malen

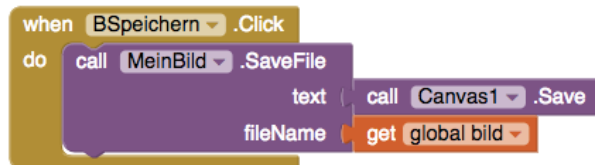


Eine mögliche Lösung um auf dem „Canvas“ zu malen



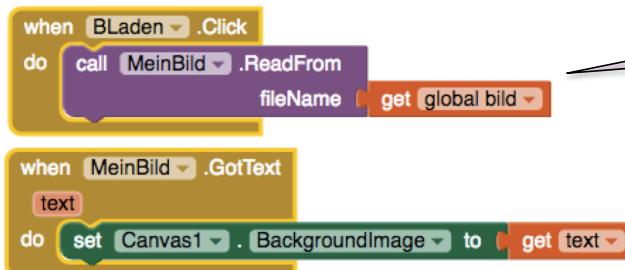
Um ein Bild zu speichern benötigst du einen Dateinamen – da du diesen an mehreren Stellen angeben musst, ist es sinnvoll eine Variable `initialize global bild to "Bild1"` zu verwenden. Das vermeidet Tippfehler und erleichtert spätere Änderungen. (der Name muss dann nur an einer Stelle geändert werden)

Um das Bild am Handy zu speichern benötigst du ein „File“. Im Beispiel wurde das „File“ mit „MeinBild“ benannt.



Die Funktion speichert den Inhalt des Canvas unter dem Dateinamen mit dem die Variable initialisiert wurde.

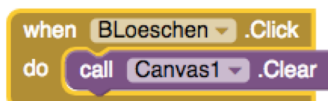
Um das gespeicherte Bild später erneut zu laden:



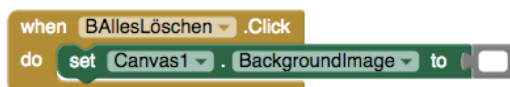
Diese Funktion liest den Inhalt der Datei, deren Name in der Variable gespeichert ist.

Ist der Lesevorgang der oberen Funktion abgeschlossen, wird der Inhalt in der lokalen Variable „text“ abgelegt und das Event MeinBild.GotText gesendet, dann wird der Hintergrund des Canvas mit dem Inhalt der lokalen Variable „text“ befüllt.

Um das „Canvas“ zu löschen.



Das ist in diesem Fall die Zeichenoberfläche und nicht der Canvas Hintergrund.
d.h. es wird nur das aktuell Gezeichnete und nicht das aus dem Speicher geladene Bild gelöscht.
Mit „Alles Löschen“ wird einfach der Hintergrund des Canvas wieder auf „weiß“ gesetzt.



Malen



Gratuliere du hast deine eigene Zeichenapp geschrieben!!

Du hast vieles von dem, was du in den vorigen Aufgaben gelernt hast angewendet und zusätzlich wieder einiges dazugelernt!

- Du hast Variablen verwendet
- Du hast ein File-Objekt verwendet um etwas dauerhaft auf deinem Gerät zu speichern
- Du hast mit Funktionen gearbeitet

Gut gemacht!!

Zusatz Malen - Speichermöglichkeit für mehrere Bilder unter verschiedenen Namen

Tipp: Speichere deine App Malen unter Male_2 nochmals ab, dann kannst du sie einfach verändern und hast beide Versionen gespeichert.

Ziel

Bei der 1. Version der App ist es nur möglich ein Bild zu speichern. Das nächste gespeicherte Bild überschreibt das vorige.

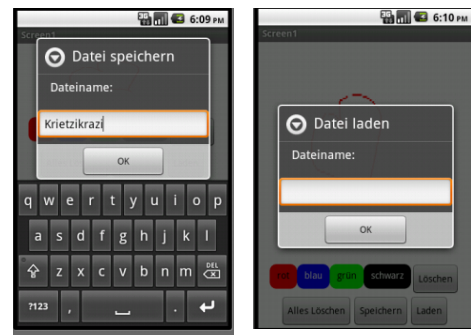
Nun sollen mehrere Bilder unter verschiedenen Namen gespeichert werden können.

Rechts siehst du das Design der App

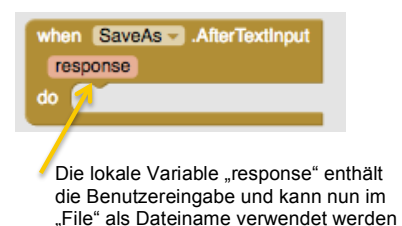
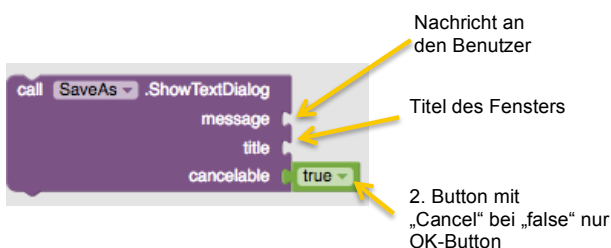
Welche Komponenten brauchst du zusätzlich dazu?

User Interface

2 x Notifier



Tipp: Mit dem Notifier hast du die Möglichkeit ein Dialogfenster (siehe oben) wie man es von vielen Anwendungen kennt zu programmieren. Da du einen Dialog zum Speichern und einen anderen zum Laden deines Bildes benötigst, brauchst du auch zwei Notifier! Im Blocks-Editor stehen dir dann für den Notifier passende Blöcke zur Verfügung. Gib auch deinen Notifiern wieder klingende Namen, damit du sie beim Programmieren leicht auseinanderhalten kannst.

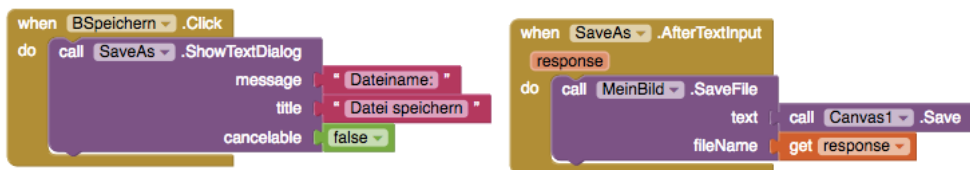


Versuche nun selbst diese Aufgabe umzusetzen.
Auf der nächsten Seite siehst du dann eine mögliche Lösung!

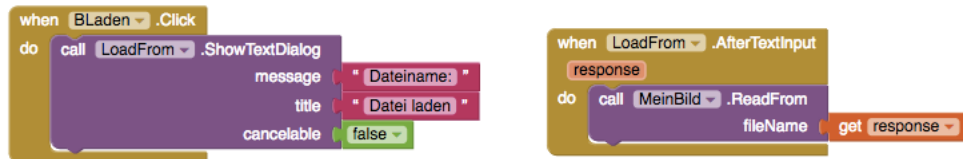
Eine von vielen möglichen Lösungen



Der **when BSpeichern.Click** Programmblock wird wie folgt verändert und ein weiterer Block kommt hinzu um die Eingabe aus dem Dialogfenster zu verarbeiten.
 In diesem Beispiel wurden die Notifier mit SaveAs bzw. LoadFrom benannt.



Der **when BLaden.Click** Programmblock wird wie folgt verändert und ein weiterer Block kommt hinzu um die Eingabe aus dem Dialogfenster zu verarbeiten.



Vergiss bitte nicht deine Anwendung zu testen!!

Geschafft!!! - Spitze

Arbeitsblatt 4 – MOLE MASH

Es wird das vielen Jugendlichen bekannte Spiel „Mole Mash“ in einer sehr vereinfachten Variante erstellt.

Im Anschluss finden die Schüler und Schülerinnen Vorschläge zur Verbesserung dieser Variante. Dazu gibt es falls diese benötigt werden auch Tipps zur Umsetzung jedoch werden keine expliziten Lösungen mehr zur Verfügung gestellt. Daher müssen sich die Schüler und Schülerinnen mit den Problemen auseinandersetzen und selbst Lösungsstrategien entwerfen.

Lernziele:

Schüler und Schülerinnen

- können Prozeduren erstellen
- können Prozeduren aufrufen
- wissen, warum es sinnvoll ist Prozeduren zu verwenden
- können mit Zufallszahlen arbeiten
- können Boolesche-Operatoren richtig einsetzen
- können Vergleichsoperatoren richtig einsetzen
- können mit mathematischen Funktionen arbeiten
- können ihren Code kommentieren
- wissen, dass es wichtig ist, ihre Projekte zu dokumentieren
- können selbstständig einen Algorithmus erstellen

Das Grundgerüst für dieses Programm wurden aus dem Video

<https://www.youtube.com/watch?v=fMayqpRLaMI> „App Inventor – Timer setzen“ übernommen.

ARBEITSBLATT 4



5.1 FANG DEN MAULWURF

Öffne bitte den App Inventor: <http://appinventor.mit.edu/explore/>
und starte ein neues Projekt: Projects -> Start new project

Ziel

Du programmierst eine sehr vereinfachte Variante des bekannten Spieles „MOLE MASH“ bzw. „FANG DEN MAULWURF“ für dein Handy

Rechts siehst du das Design der App

Welche Komponenten brauchst du dazu?

User Interface

1 x TextBox

Layout

1 x HorizontalArrangement

Drawing and Animation

1 x Canvas

1 x ImageSprite

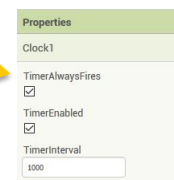
Sensors

1 x Clock

Media

1 x Maulwurf.png

Bilddateien für diese App findest du auf
https://drive.google.com/Bilder_Töne



Wie kannst du das Ziel erreichen?

Die TextBox platzierst du mit Hilfe des „TableArrangements“ auf dem virtuellen Bildschirm.

Tipp: Im Video wurden nicht alle Komponenten umbenannt – klingende Bezeichnungen erleichtern jedoch das Arbeiten

Immer wenn der „Timer“ abgelaufen ist, soll der Maulwurf auf eine zufällige Position am Bildschirm gesetzt werden.



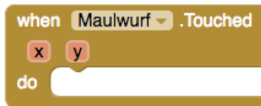
Du musst die Werte so wählen, dass der Maulwurf den Bereich des Bildschirms nutzen kann

Tipp: Wenn du den Maulwurf im Designer auf dem Bildschirm an einer beliebigen Stelle platzierst, kannst du in den „Properties“ die Koordinaten seiner Position ablesen

zu 4.1 FANG DEN MAULWURF einfache Version



Wenn der Maulwurf gefangen wird, willst du, dass sich der „Score um 1 verändert.



Du benötigst eine globale Variable, in welcher du den aktuellen Spielstand speichern kannst. Als erstes musst du sie mit einem Namen und einem Startwert versehen (d.h. du musst sie initialisieren)



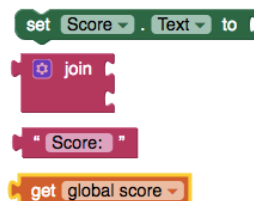
Deine Variable braucht einen Namen und einen Startwert

Wenn der Maulwurf gefangen wurde, wird der Zähler um 1 erhöht. Diese Blöcke kannst du dafür verwenden. Überlege an welcher Stelle des Programms diese Blöcke eingefügt werden müssen.



So kannst du auf deine Variable zugreifen

Damit dein Zählerstand auch am Bildschirm angezeigt wird:



Mit „join“ kannst du Texte zusammenfügen

Score ist hier der Name der Komponente in welcher du den Zählerstand ausgibst

Vergiss bitte nicht, deine App zu testen!

Aufgabe: Du hast beim Test sicher festgestellt, dass Einiges in dieser Anwendung noch nicht optimal gelöst ist und Manches fehlt.

Verbessere das Spiel!! Auf der nächsten Seite findest du einige Anregungen und Tipps.

Kommentiere deine Befehle – um es später nachvollziehen zu können.

Dokumentiere dein Programm und notiere wo du Schwierigkeiten hattest.

4.2 FANG DEN MAULWURF Verbesserungen



1. Das Spiel startet sofort. Mit einem Startknopf könntest du entscheiden, wann das Spiel beginnt
2. Das Spiel hat kein Ende. Ein mögliches Ende wäre, wenn du auf den Bildschirm klickst und dabei den Maulwurf verfehlst.
3. Der **Zähler** sollte bei einem Neustart wieder **auf Null** gesetzt werden.
4. Ein „**Highscore**“ sollte den bisher höchsten erreichten Punktestand anzeigen.
5. Ein alternatives Spielende: Du hast nur eine bestimmte Zeit zur Verfügung und das Spiel endet sobald die Zeit abgelaufen ist.
6. Das Spiel könnte etwas schwieriger werden. Zum Beispiel wäre es möglich eine weitere Spielfigur einzubauen. Fängt man diese zweite Figur werden Punkte abgezogen.

Aufgabe: Versuche diese Anforderungen eigenständig zu implementieren. Solltest du etwas Hilfe benötigen, findest du auf der nächsten Seite einige Hinweise.

Viel Spaß!

So ähnlich könnte deine verbesserte App aussehen:



Tipp: Halte dir immer im Bewusstsein, dass es sich bei einem **Algorithmus** um eine Schritt-für Schritt Handlungsanweisung für deinen Computer handelt. Die Einzelbefehle werden in der von dir vorgegebenen Reihenfolge abgearbeitet.

Lass, während du programmierst, die App am Handy oder am Emulator laufen – das „Live-Testing“ des App Inventors ermöglicht es, dass du jede kleine Änderung unmittelbar überprüfen kannst.

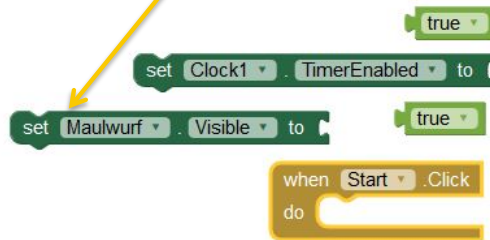
4.3 Hilfestellung Startknopf (Punkt 1.)

Um zu verhindern, dass der Maulwurf sofort nach dem Öffnen der App herumspringt musst du im „Designer“ die „Properties“ der Komponente „Clock“ ändern.

Du kannst hier das Feuern des Timers ausschalten, indem du das Häkchen bei „TimerEnabled“ entfernst. Willst du vor dem Start den Maulwurf nicht sehen, entferne in den „Properties“ des Maulwurfs das Häkchen bei „Visible“.

Damit du das Spiel aktiv starten kannst, benötigst du erst einmal einen Start Button. Diesen kannst du wie gewohnt im „Designer“ hinzufügen.

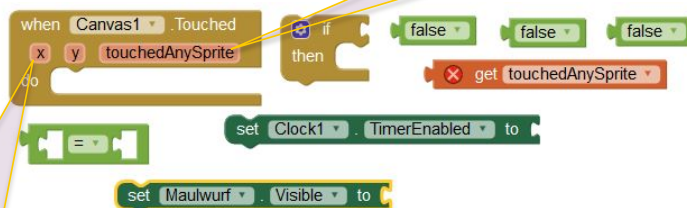
Wenn nun der Startbutton angeklickt wird muss der **Maulwurf** wieder **sichtbar** und der **Timer** wieder auf „Enabled“ geschaltet werden.



4.4 Hilfestellung Spiel-Ende Variante 1 (Punkt 2.)

Das Spiel endet, wenn du das „Canvas“ und nicht den Maulwurf berührst.

Wenn das „Canvas“ berührt wird:

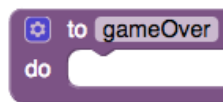


Es werden die Parameter des Berührungspunktes also x- und y-Koordinaten

Falls dabei keine Spielfigur berührt wurde, wird der **Maulwurf** auf **nicht sichtbar** und der **Timer** auf „nicht Enabled“ geschaltet.

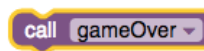
zu 4.4 gameOver Prozedur (Punkt 2.)

Tipp: Da in Punkt 5 ein weiteres Spielende vorgesehen ist, wäre es äußerst sinnvoll sämtliche Befehle, welche am Ende des Spiels ausgeführt werden sollen in eine **Prozedur** „gameOver“ zu geben. Das ermöglicht es dir „gameOver“ an mehreren Stellen deines Programmes zu verwenden. Du musst dann nicht immer alle notwendigen Einzelbefehle eingeben, sondern nur die Prozedur aufrufen. Wenn du später etwas am Spielende ändern willst, brauchst du das nur an einer Stelle im Programm machen. Das erleichtert die Arbeit ☺.

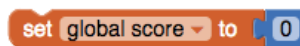


Der **Maulwurf** wird immer auf **nicht sichtbar** und der **Timer** auf „**nicht Enabled**“ geschaltet, ganz egal wodurch das Spielende ausgelöst wird.

Wenn das „Canvas“ berührt wird muss die Bedingung abgefragt werden. Das heißt falls keine Spielfigur berührt wurde wird die Prozedur „gameOver“ mit folgendem Befehl aufgerufen.



4.5 Hilfestellung Punktestand bei Neustart auf Null (Punkt 3.)



Wenn der Startknopf geklickt wird, wird auch der Score wieder auf Null gesetzt.

4.6 Hilfestellung zur Anzeige des Highscores (Punkt 4.)

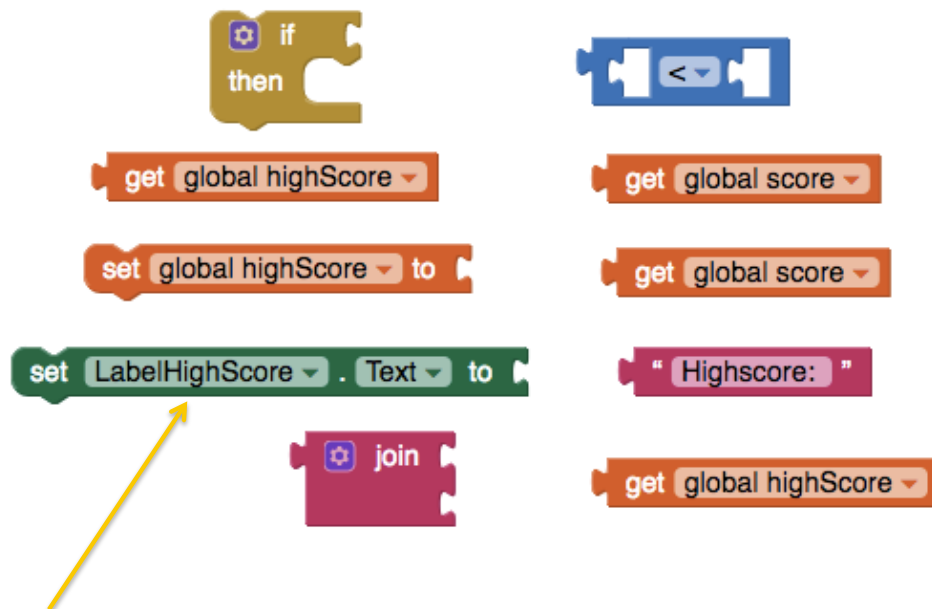


Du brauchst ein Textfeld oder ein Label um den Highscore anzeigen zu können. Dieses platzierst du im „Designer“ an eine Stelle deiner Wahl.

Um den höchsten bisher erreichten Punktestand zu speichern benötigst du eine weitere **Variable**. Auch diese musst du wieder mit einem Namen und einem Startwert versehen.

initialize global highScore to 0

Falls der aktuell gespeicherte Wert der Variable „highScore“ kleiner als der aktuelle Wert in der Variable „score“ ist, wird die Variable „highScore“ auf den Wert der Variable „score“ gesetzt.



Der Text im Feld „HighScore“ wird auf „Highscore: und den Wert, welcher in der Variable highScore gespeichert ist“ gesetzt.

Überlege, dass dies alles dann geschehen muss, wenn dein aktuelles Spiel beendet wird.

Tipp: Den Programmblock den du gerade für deinen Highscore geschrieben hast, kannst du zu deiner Prozedur „gameOver“ hinzufügen, da ja der Highscore auch immer bei Spielende aktualisiert werden soll.

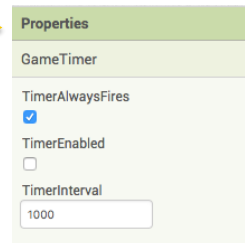
4.7 Spielende nach abgelaufener Zeit (Punkt 5.)



Du brauchst eine zweite „Clock“ um dieses Spielende zu programmieren.

Diese fügst du im „Designer“ hinzu. Du findest sie unter „Sensors“

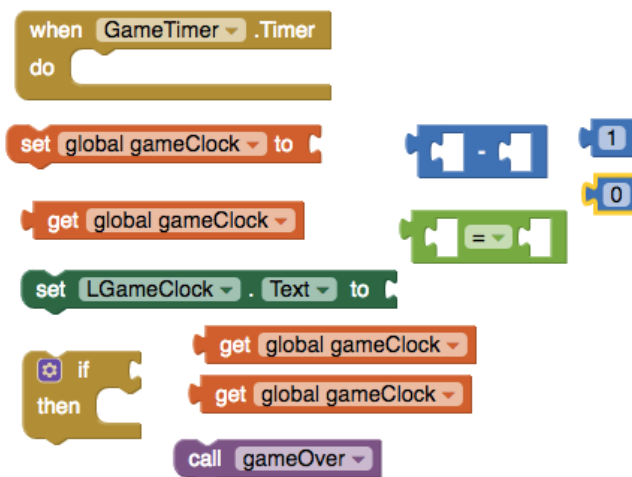
Du brauchst auch eine **Variable** um den aktuellen Wert des Timers speichern zu können



initialize global gameClock to 60

Startwert 60 bedeutet, dass du 60 Sekunden lang spielen kannst. Du kannst jeden beliebigen Wert verwenden.

Ebenso brauchst du ein **Textfeld** im dem du anzeigen kannst, wie viele Sekunden noch zur Verfügung stehen



Wenn der **GameTimer** (so wurde die zweite Clock im Beispiel benannt) feuert, wird die globale Variable „gameClock“ um 1 verringert und der Text im Textfeld „LGameClock“ auf den aktuellen Inhalt der globalen Variable „gameClock“ gesetzt.

Falls dabei der aktuell gespeicherte Wert in der globalen Variable „gameClock“ den Wert Null erreicht, rufe die Prozedur „gameOver“ auf.

Bei jedem Neustart des Spieles muss die globale Variable „gameClock“ erneut den Startwert 60 zugewiesen bekommen, da sie ja bereits nach dem ersten Mal spielen den Wert Null gespeichert hat. Du musst also, wenn der Startbutton geklickt wird auch folgenden Befehl einbauen.

set global gameClock to 60

Vergiss bitte nicht, deine App zu testen!

4.8 Hilfestellung - Das Spiel wird schwieriger" (Punkt 6.)



Um diese Variante zu realisieren, musst du erst mal ein weiteres Bild von einer Spielfigur im Designer hinzufügen.
Dieses kannst du unter https://drive.google.com/Bilder_Töne herunterladen oder verwende eines deiner Bilder.
Im Beispiel wurde ein Eichhörnchen (Squirrel) verwendet.

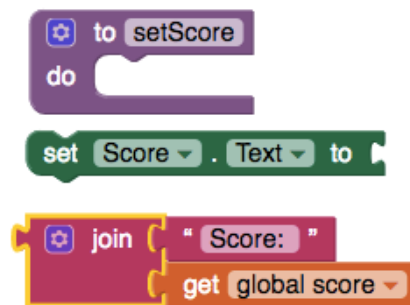
Es ist auch notwendig bereits geschriebene Funktionen wieder abzuändern.

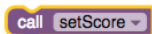
Was brauchst du?

Du brauchst eine Prozedur die den Maulwurf auf dem Bildschirm positioniert
Du brauchst eine Prozedur die das Eichhörnchen auf dem Bildschirm positioniert

Da du aber nicht möchtest dass beide Spielfiguren gleichzeitig auf dem Bildschirm auftauchen, brauchen wir eine Prozedur die entscheidet, welche der beiden oberen Prozeduren aufgerufen wird. Die Entscheidung darüber soll zufällig sein.

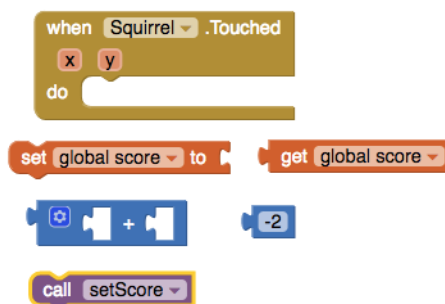
Der Punktestand verändert sich immer wenn du eine Spielfigur berührst. Das heißt, dass die Anzeige des Punktestandes sich beim Neustart, beim Berühren des Eichhörnchens und beim Berühren des Maulwurfs ändert. Daher ist es sehr sinnvoll diese Befehle in einer Prozedur zusammenzufassen, da du sonst den gleichen Code immer und immer wieder schreiben müsstest.



Diese Prozedur kannst du dann an der benötigten Stelle mit  aufrufen.

Du kannst das gleich einmal beim **Maulwurf** ausprobieren.

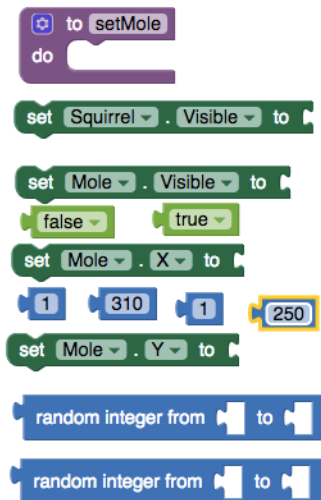
Für das Eichhörnchen brauchst du einen ganz ähnlichen Block wie für den **Maulwurf**
Der Unterschied ist nur, dass diesmal 2 Punkte abgezogen werden, wenn das Eichhörnchen berührt wird.



zu 4.8 Hilfestellung - Das Spiel wird schwieriger" (Punkt 6.)

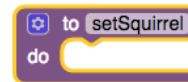


In der Prozedur zum Platzieren des Maulwurfes brauchst du alle Befehle die du Anfangs im Block **when Clock1.Timer** verwendet hast und auch jene Befehle aus dem Startblock welche nur den Maulwurf betreffen.

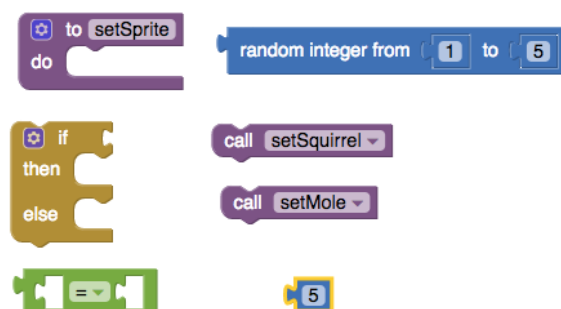


Zusätzlich musst du die Sichtbarkeit des Eichhörnchens ausschalten

Die Prozedur für das Eichhörnchen sieht fast gleich aus. In diesem Fall musst du aber den Maulwurf auf Unsichtbar schalten



Jetzt brauchst du nur noch eine Prozedur, welche sich zwischen Maulwurf und Eichhörnchen entscheidet.

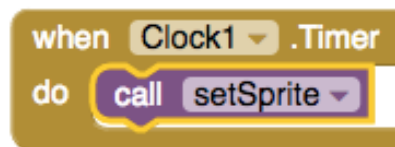


Die Prozedur **setSprite** ruft die Prozedur **setSquirrel** auf falls die zufällig gewählte Zahl aus 1 bis 5 genau 5 ist. Ansonsten wird die Prozedur **setMole** aufgerufen.

zu 4.8 Hilfestellung - Das Spiel wird schwieriger" (Punkt 6.)



Fast geschafft!!! Jetzt muss die Prozedur setSprite nur noch aufgerufen werden, sobald der Timer der ersten Uhr feuert.



Du hast selbst ein Spiel programmiert – GLÜCKWUNSCH!!!!

- Du hast mit Zufallszahlen gearbeitet
- Du hast Vergleichsoperatoren verwendet
- Du hast mathematische Funktionen verwendet
- Du hast mit Prozeduren gearbeitet
- Du hast gelernt, dass man Codeblöcke wiederverwenden kann und soll
- Du hast mit Booleschen-Operatoren gearbeitet (true, false)
- Du hast mit Bedingungen gearbeitet und den Befehl if-then-else verwendet
- Du hast gelernt, dass es wichtig ist seinen Code zu kommentieren

VIEL SPASS!!!

Ganz bestimmt fallen dir noch ein paar Verbesserungen für dieses Spiel ein.

Probiers einfach aus!!!!

Arbeitsblatt 5 – PONG

In diesem Beispiel wird zu Beginn ein Grundgerüst für das Spiel PONG erstellt. Zu dieser sehr einfachen Variante ist auch ein Erklärvideo in den Unterlagen zu finden.

Im weiteren Verlauf werden einige Verbesserungsvorschläge gemacht. Sollte die Umsetzung Schwierigkeiten bereiten, finden die Schüler und Schülerinnen auf den folgenden Seiten eine Hilfestellung. Es wird allerdings nicht mehr der zusammengesetzte Code vorgegeben, sondern nur noch die zu verwendenden Blöcke gezeigt und beschrieben.

Lernziele:

Schüler und Schülerinnen

- können mit Zufallszahlen arbeiten
- können Boolesche-Operatoren richtig einsetzen
- können Vergleichsoperatoren richtig einsetzen
- können mit mathematischen Funktionen arbeiten
- können ihren Code kommentieren
- wissen, dass es wichtig ist, ihre Projekte zu dokumentieren
- können selbstständig einen Algorithmus erstellen

Das Grundgerüst für dieses Spiel wurde aus dem Video

<https://www.youtube.com/watch?v=v4NfNFO1r9E&t=130s> „App Inventor – PONG“ übernommen.

A R B E I T S B L A T T 5



5.1 PONG einfache Version

Öffne bitte den App Inventor: <http://appinventor.mit.edu/explore/>
und starte ein neues Projekt: Projects -> Start new project

Ziel

Du programmierst eine sehr vereinfachte Variante des Spiele „PONG“ für dein Handy

Rechts siehst du das Design der App

Welche Komponenten brauchst du dazu?

User Interface

1 x Button
1 x Label

Layout

1 x TableArrangement

Drawing and Animation

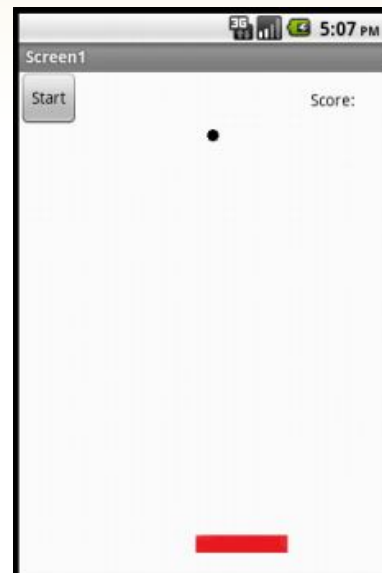
1 x Canvas
1 x Ball
1 x ImageSprite

Media

1 x balken.png

Bilddateien für diese App findest du auf
https://drive.google.com/Bilder_Töne

Ein Anleitungsvideo findest du auf der Lernplattform

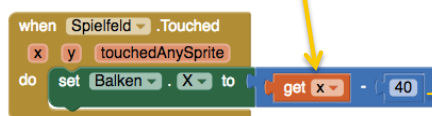
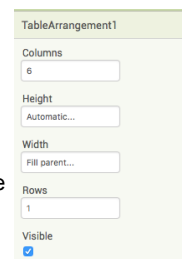


Wie kannst du das Ziel erreichen?

Den Button und das Label platzierst du mit Hilfe des TableArrangements auf dem virtuellen Bildschirm.

Tipp: Im Video wurden nicht alle Komponenten umbenannt – klingende Bezeichnungen erleichtern jedoch das Arbeiten

Wenn das „Canvas“ bzw. das Spielfeld berührt wird, soll der Balken auf die entsprechende x-Koordinate gesetzt werden. (hier noch minus 40, damit die Mitte des Balkens auf die gewünschte Position gesetzt wird und nicht der Rand des Balkens)



Die y-Koordinate des Balkes soll nicht verändert werden - nur der x-Wert wird verwendet

zu 5.1 PONG einfache Version

when Start .Click

do

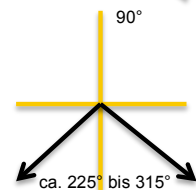
- set Ball1 .Heading to random integer from 225 to 315
- set Ball1 .Speed to 5
- set Ball1 .Interval to 10

Startrichtung des Balles mittels Zufallszahl (in einem Winkel zwischen 225° und 315°)

Geschwindigkeit des Balles

Intervall in dem die Position neu gezeichnet wird

Linke Maustaste am Block öffnet eine Liste – wähle Add Comment, um diese Blöcke zu kommentieren



Bei Kollision mit dem Balken (other = Balken) - setze die Richtung des Balles auf 360 Grad minus der aktuellen Richtung des Balles (=entgegengesetzt)

when Ball1 .CollidedWith

other

do

- set other to Balken
- set Ball1 .Heading to $360 - \text{Ball1} .\text{Heading}$

da eine Kollision nur mit dem Balken möglich ist (es gibt keine anderen Objekte) – bräuchte man diesen Befehl nicht

Bei der Kollision des Balles mit dem Balken, bewegt sich der Ball in die entgegengesetzte Richtung (360° minus der momentanen Richtung des Balles = entgegengesetzte Richtung)

when Ball1 .EdgeReached

edge

do

- if $\text{get edge} = -1$
 - then set Zähler .Text to "Game Over"
- if $\text{get edge} \neq -1$
 - then call Ball1 .Bounce
 - edge get edge

Berührt der Ball die untere Kante wird der Text des Zählers auf „Game Over“ gesetzt

Berührt der Ball nicht die untere Kante soll er von der Kante die er berührt abprallen

Die untere Kante des Bildschirms ist -1 im App Inventor

Anstelle der beiden if-then Blöcke, wäre ein if-then-else Block eleganter ☺ VERSUCHS DOCH MAL!!!



Vergiss bitte nicht, deine App zu testen!

Aufgabe: Du hast beim Test sicher festgestellt, dass Einiges in dieser Anwendung noch nicht optimal gelöst ist.

Verbessere das Spiel!! Auf der nächsten Seite findest du einige Anregungen und Tipps.

Kommentiere deine Befehle – um es später nachvollziehen zu können.

Dokumentiere dein Programm und notiere wo du Schwierigkeiten hattest.

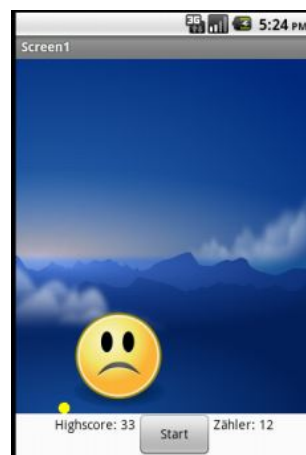
5.2 PONG Verbesserungen

1. Das **Feld „Zähler“** sollte den aktuellen Punktestand anzeigen.
2. Der **Zähler** sollte bei einem Neustart wieder **auf Null** gesetzt werden.
3. Ein „**Highscore**“ sollte den bisher höchsten erreichten Punktestand anzeigen.
4. Der Ball soll die Bewegung stoppen wenn er neben den Balken fällt
5. Die **Bewegung des Balkens** sollte mit „**wischen**“ am Bildschirm möglich sein.
6. Das Spiel soll mit zunehmender Dauer schwieriger werden. (z.B. der Ball wird nach jeweils 5 Berührungen mit dem Balken schneller – oder/und der Balken wird nach jeweils 15 Berührungen mit dem Ball kleiner – oder/und der Balken bewegt sich nach jeweils 15 Berührungen mit dem Ball nach oben)
7. Die **Spielefläche** soll ein **Bild** zeigen, welches sich bei jedem neuen Level (Balken wurde kleiner bzw. höher) ändert.
8. Du hast bestimmt noch einige weitere gute Ideen, wie das Spiel noch verbessert werden könnte – versuche auch diese umzusetzen. (Besprich dein Vorhaben kurz mit deiner Lehrperson oder in der Gruppe)

Aufgabe: Versuche diese Anforderungen eigenständig zu implementieren. Solltest du etwas Hilfe benötigen, findest du auf der nächsten Seite einige Hinweise.

Viel Spaß!

So ähnlich könnte deine verbesserte App aussehen:



Tipp: Halte dir immer im Bewusstsein, dass es sich bei einem **Algorithmus** um eine Schritt-für Schritt Handlungsanweisung für deinen Computer handelt. Die Einzelbefehle werden in der von dir vorgegebenen Reihenfolge abgearbeitet.

Lass während du programmierst die App am Handy oder am Emulator laufen – das „Live-Testing“ des App Inventors ermöglicht es, dass du jede kleine Änderung unmittelbar überprüfen kannst.

5.3 PONG eine kleine Hilfestellung zum Zähler (Punkt 1.)



initialize global score to

Deine Variable braucht einen Namen und einen Startwert

Du benötigst eine globale Variable, in welcher du den aktuellen Spielstand speichern kannst. Als erstes musst du sie mit einem Namen und einem Startwert versehen (d.h. du musst sie initialisieren)

Wenn der Ball den Balken berührt wird der Zähler um 1 erhöht. Diese Blöcke kannst du dafür verwenden. Überlege an welcher Stelle des Programms diese Blöcke eingefügt werden müssen.

set global score to

+

get global score

So kannst du auf deine Variable zugreifen

Damit dein Zählerstand auch am Bildschirm angezeigt wird:

set Zähler Text to

join " Zähler: "

get global score

Zähler ist hier der Name der Komponente in welcher du den Zählerstand ausgibst

Mit „join“ kannst du Texte zusammenfügen

5.4 Hilfestellung Zählerstand beim Start auf Null (Punkt 2.)

Um den Zählerstand bei einem Neustart wieder auf Null zu setzten. Überlege auch hier wo genau im Programm dieser Befehl eingebaut werden muss

set global score to 0

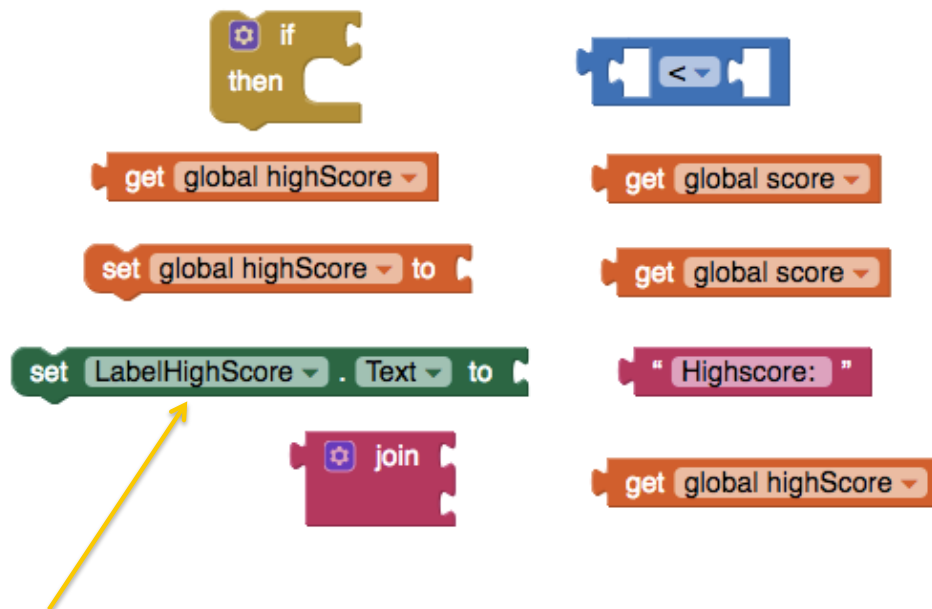
5.5 Hilfestellung zur Anzeige des Highscores (Punkt 3.)



Um den höchsten bisher erreichten Punktestand zu speichern benötigst du eine weitere Variable. Auch diese musst du wieder mit einem Namen und einem Startwert versehen.

initialize global highScore to 0

Ist der aktuell gespeicherte Wert der Variable „highScore“ kleiner als der aktuelle Wert in der Variable „score“, wird die Variable „highScore“ auf den Wert der Variable „score“ gesetzt.



Der Text im Feld „HighScore“ (welches du vorher im Designer auf dem Bildschirm platzieren musst) wird auf „Highscore: und dem Wert, welcher in der Variable highScore gespeichert ist“ gesetzt.

Überlege, dass dies alles dann geschehen muss, wenn der Ball die untere Kante erreicht. Diese Überlegung wird dir helfe herauszufinden, an welcher Stelle im Programm diese Blöcke untergebracht werden müssen.

5.6 Ball stoppt wenn er neben den Balken fällt (Punkt 4.)

set Ball1 . Enabled to false

Dem Ball muss dann natürlich beim Start wieder in Bewegung gesetzt werden:

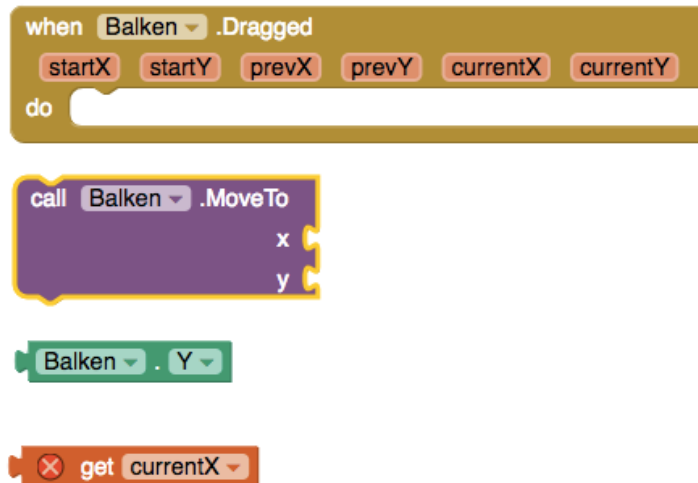
set Ball1 . Enabled to true

4.7 Hilfestellung – Balken mit “wischen bewegen” (Punkt



Da du jetzt nicht mehr willst, dass sich der Balken nur bewegt, wenn du den Bildschirm erneut berührst, kannst du den Block „when Spielfeld.Touched“ löschen.

Nun brauchst du einen Block der auf „wischen“ über den Bildschirm reagiert.

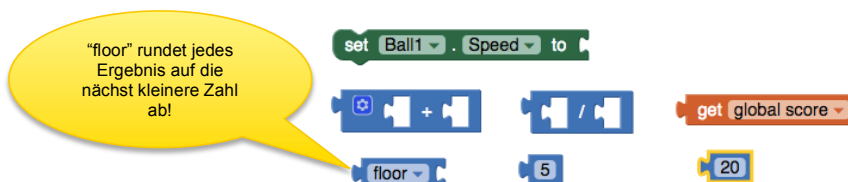


Die lokale Variable „currentX“ hat den Wert der X-Koordinate, auf welcher sich der Finger bzw. der Mauszeiger gerade befindet als Inhalt. Dieser Inhalt muss der Funktion „call Balken.MoveTo“ übergeben werden, damit sich der Balken in die richtige Position begibt.

Als Y-Koordinate erhält die Funktion den „Y-Startwert“ des Balkens, da er sich ja lediglich nach rechts und links bewegen können soll.

5.8 Hilfestellung – Das Spiel wird schwieriger” (Punkt 6.)

Variante 1: Der Ball soll immer nach 20 Berührungen mit dem Balken seine Geschwindigkeit um 1 erhöhen.



Die Geschwindigkeit des Balles wird jeweils nach 20 Berührungen um 1 erhöht, wenn du den aktuellen Score durch 20 dividierst und 5 addierst und das Ergebnis abrundest.

Beispiel: (Score = 10) -> $(10/20) + 5 = 5$ abgerundet
(Score = 21) -> $(21/20) + 5 = 6$ abgerundet

zu 5.8 Hilfestellung – Das Spiel wird schwieriger” (Punkt



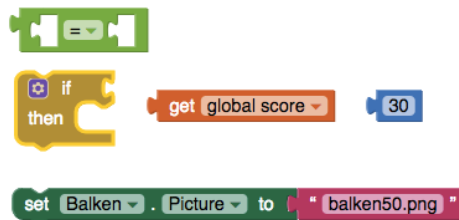
Variante 2: Der Balken soll immer nach 20 Berührungen mit dem Balle kleiner werden.

Um diese Variante zu realisieren, musst du erst mal weitere Bilder von Balken im Designer hinzufügen. Diese kannst du unter https://drive.google.com/Bilder_Töne herunterladen.



Beim Start hast du dem Balken ein Bild zugewiesen. Nun sollst du dieses Bild ändern, sobald der Score 15 ist.

Dies kannst du natürlich beliebig oft machen. Also zum Beispiel ein noch kleineres Bild dem Balken zuweisen, sobald der Score 30 ist.



5.9 Hilfestellung – optische Veränderungen (Punkt 7.)

Ähnlich wie im Punkt 6 kannst du auch dem „Canvas“ ein Bild zuweisen und eine neue Zuweisung vornehmen, sobald ein gewisser Punktestand erreicht ist.

Du ein eigenes Spiel programmiert und dabei viel gelernt und hoffentlich auch Spaß gehabt.

- Du hast mit Zufallszahlen gearbeitet
- Du hast die Vergleichsoperatoren verwendet
- Du hast mathematische Funktionen verwendet
- Du hast mit Booleschen-Operatoren gearbeitet (true, false)
- Du hast mit Bedingungen gearbeitet und den Befehl if-then-else verwendet
- Du hast gelernt, dass es wichtig ist seinen Code zu kommentieren

GRATULATION UND VIEL VERGNÜGEN MIT DEINEM SELBST ERSTELLTEN SPIEL!!!

Videos erstellen

Die Lehrperson hat die Möglichkeit auf eine Vielzahl vorhandener Videos zurückzugreifen.

[GoogleAccount_anlegen_löschen](#)

[Einführung App Inventor I](#)

[Einführung App Inventor II](#)

[Video_Prozeduren](#)

[Video_Sensoren](#)

[Video_Maulwurfspiel \(Timer\)](#)

[Video_Spiel Pong](#)

Bei der Verwendung fremder Videos ist jedoch immer auf die Rechte zu achten, insbesondere ist die Änderung und Wiederveröffentlichung oft nicht gestattet. Hierzu muss man unbedingt auf die Urheberrechte achten und überprüfen, ob es sich um Open Content handelt.

Da Videos aus fremder Produktion nicht immer genau zum eigenen Unterricht passen, oder nicht kohärent in Schreibweise oder Bezeichnungen sind, wird man möglicherweise nicht umhinkommen, seine eigenen Videos zu erstellen.

Die Produktion von Erklärvideos ist kostengünstig und relativ einfach. Man benötigt einen Tablett PC, einen Grafikstift, ein Mikrofon und eventuell eine Webcam.

Ebenfalls benötigt wird ein Screencastprogramm wie:

- sceencast-o-matic (kostenfrei, Notizfunktion, gebrandet)
- camtasia www.techsmith.de/camtasia.html (kostenpflichtig)
- Jing www.techsmith.com/jing.html (max. 5 min Aufnahmezeit)
- VLC Media Player (kostenlos)
- Camstudio
- Quicktime Player (Mac)
- Office Mix (Add-In zu Power-Point, kostenlos)
- www.imore.com/best-screen-redording (für MAC)
- showme.com (für den Einsatz von Tablets)
- educanon.com (Videos schneiden und mit Fragen versehen, bzw. bestehende Videos anpassen)

Bei der Erstellung von Erklärvideos ist zu beachten, dass diese nicht zu lange werden und, dass sie nicht qualitativ hochwertig sein müssen (abgesehen vom Inhalt). Sie dienen lediglich als Medium zur Stoffvermittlung und sollen den Schülerinnen und Schülern helfen die Aufgaben zu lösen.

Videos bei denen hohe Professionalität angestrebt wird, wirken oft spröde und mitunter langweilig. (Christian Spannagel, 06:12:55 UTC)

„At some point a teacher must decide if a video needs to be perfect, or if it is needed on Tuesday”

(Jonathan Bergmann and Aaron Sams, 2012)

Selman Khan gilt als Vorbild. Man spricht von „Khan-Style-Videos“. Für ihn sind folgende vier Punkte relevant:

- „*Knowledge Café*“: Der Tonfall ist das Wichtigste.
Es soll eine authentische Person auf Augenhöhe mit dem Zuhörer sprechen und die Stimme muss emotional bewegt sein. Kein Dozieren, nicht von oben herab, sondern persönlich und intim. Beim Lernenden sollte ein „Selbstgespräch“ angeregt werden.
- „*Visuell und farbig*“: Khan-Style-Videos sind minimalistisch.
Es wird mit unterschiedlichen Farben auf schwarzem Hintergrund geschrieben und gezeichnet. Denn erst wenn es handgemacht und „unperfekt“ ist wird es zum Denkwerkzeug.
- „*Sich vorbereiten*“: Khan rät zur gedanklichen Vorbereitung ohne Skript (lediglich Stichworte) und zum Sprechen in einem Zug.
- „*So kurz wie möglich, aber nicht zu kurz*“: Nur einen Gedankenbogen pro Video. Lektionen mit circa 6 – 8 Minuten Länge schaffen einen Rhythmus. (Lindner, 2013)

Ebenfalls sollten die Prinzipien von Richard E. Mayer zur kognitiven Theorie des multimedialen Lernens beachtet werden.

- „*Redundanzprinzip*“: Bild, Ton und Text sollten nicht die gleichen Informationen geben, sondern sich gegenseitig ergänzen bzw. unterschiedliche Blickwinkel eröffnen.
- „*Multimediaprinzip*“: bessere Ergebnisse werden erzielt, wenn einem Text geeigneter Bilder hinzugefügt werden.
- „*Segmentierungsprinzip*“: Aufteilung in Abschnitte – Lernender wird durch Unterbrechungen aktiviert
- „*Prinzip des Vorwissens*“: Wesentliche Begriffe, Bezeichnungen etc. schon vorab einführen.
- „*Modalitätsprinzip*“: Gesprochener Text zu Abbildungen oder Grafiken
- „*Kontiguitätsprinzip*“: räumliche Nähe zwischen Text und Visualisierung um die kognitive Belastung gering zu halten, sowie die zeitgleiche zur Verfügung stellen von Erklärung und Animation.
- „*Kohärenzprinzip*“: keine „verführerischen“ Texte und Illustrationen, keine Klanguntermalung – alles Irrelevante streichen, um die Ressourcen des Arbeitsgedächtnisses zur Gänze für die wichtigen Inhalte bereit zu haben
- „*Signalisierungsprinzip*“: Hinweiszeichen zur Hervorhebung des Kerninhaltes
- „*Personalisierungsprinzip*“: Formulierung in umgangssprachlicher Form führen zu besseren Lernleistungen als formale Ausdrücke. Der Lernende sollte möglichst persönlich angesprochen werden („du“ statt „man“). Die Emotionen des Vortragenden sollten spürbar sein, somit wird eine soziale Reaktion beim Hörenden ausgelöst und das führt zu einer Steigerung der aktiven, kognitiven Verarbeitung, was wiederum die Verständnisleistung verbessert.
- „*Stimmprinzip*“: Keine computergenerierten Stimmen verwenden.
- „*Bildprinzip*“: Es wird nicht besser gelernt, wenn der Sprecher auch zu sehen ist
(*Multimedia-Prinzipien vom Mayer (2009)* (Martin Ebner, Sandra Schön, 2013))

Google Account

GOOGLE ACCOUNT

Account erstellen

Öffne bitte die Webseite von Google: <https://www.google.at>

- Klicke auf den Button "Anmelden". Dieser befindet sich auf der rechten oberen Seite
- Klicke dann unten auf "Konto erstellen" – du kommst zum Formular für die Registrierung
- Gib alle geforderten Daten ein. Beachte, dass der gewählte Name als Anzeige-Name dient
- Du hast die Wahl deine bestehende E-Mail-Adresse zu verwenden, oder eine neue Gmail-Adresse zu erstellen
- wähle den Button "Nächster Schritt".
- Scrolle durch die Datenschutzerklärung und bestätige diese mit "Ich stimme zu".
- Dein Google-Konto wird nun erstellt.
- Wenn du eine alternative E-Mail-Adresse eingegeben hast, kann es sein, dass du diese noch bestätigen musst.

Hinweis: Du musst weder eine Telefonnummer noch eine alternative E-Mailadresse angeben. Vergisst du aber dein Passwort, können diese Daten dir weiterhelfen.

Account löschen

Öffne bitte die Webseite von Google: <https://www.google.at>

- Melde dich mit deinem Account an
- Klicke rechts oben auf dein Profilbild bzw. dein Profilzeichen und wähle „Mein Konto“
- Scrolle auf der Seite ganz nach unten bis zu den Kontoeinstellungen und wähle „Konto oder Dienste löschen“
- Wähle „Google-Konto und Daten löschen“
- Du musst nun nochmals das Passwort eingeben um deine Identität zu bestätigen. Klicke anschließend auf „Weiter“
- Setze beide Häkchen und stimme zu
- Nun ist dein Konto wieder erfolgreich gelöscht!

Im Google-Konto und alle zugehörigen Daten wurden gelöscht

Wenn das Google-Konto vollständig gelöscht haben, haben Sie folgende Schritte zu befolgen:

1. Prüfen Sie das Löschen auf
2. Prüfen Sie, ob alle Daten gelöscht sind
3. Prüfen Sie, ob alle Daten gelöscht sind

Quellen und verwendete Literatur

Abbildungsverzeichnis

Abbildung 2 - Testen am Handy mit WLAN	14
Abbildung 3 -Companion App	14
Abbildung 4 – Verbindungsfenster	14
Abbildung 5 - QR-Verbindungscode	14
Abbildung 6 - Testen am Computer	15
Abbildung 7 - Testen am Handy mit USB-Verbindung	15
Abbildung 9 – Start- bzw. Designfenster	17
Abbildung 11 - Blocks Editor	20

Quellen:

App Inventor – Pong <https://www.youtube.com/watch?v=v4NfNFO1r9E>

App Inventor – Timer setzen <https://www.youtube.com/watch?v=fMayqpRLaMI>

Christian Spannagel. *Der Flipped Classroom - Slides reloaded*. Bildung. Abgerufen von <https://de.slideshare.net/cspannagel/der-flipped-classroom-slides-reloaded> (last visited 05.04.2018)

Einrichten App Inventor | Entdecken MIT App Inventor. . Abgerufen 27. Februar 2018, von <http://appinventor.mit.edu/explore/ai2/setup.html>

Jonathan Bergmann and Aaron Sams. (2012). *Flip your Classroom - reach Every Student in Every Class Every Day*.

Lindner, M. (2013, Juli 2). Wie macht man MOOC-Videos im Khan-Style? Abgerufen 2. März 2018, von <http://mathemooc.de/2013/07/02/wie-macht-man-mooc-videos-im-khan-style/>

Martin Ebner, Sandra Schön. (2013). *Lehrbuch für Lernen und Lehren mit Technologien*.

Motschnig-Pitrik, R., Kabicher, S. & Figl, K Didaktische und kommunikative Aspekte. *ZML Innovative Lernszenarien*, S. 73.

Rollke, Karl-Hermann, *Android Apps mit Appinventor2: Jeder kann programmieren*, CreateSpace Independent Publishing Platform 2017, ISBN-13: 978-1544256009

Setting Up App Inventor | Explore MIT App Inventor. Abgerufen 1. März 2018, von <http://appinventor.mit.edu/explore/ai2/setup.html>

Bildnachweise:

Abbildung 2,6,7:	http://appinventor.mit.edu/explore/ai2/setup.html
Besprechung:	https://pixabay.com/de/besprechung-meeting-gespr%C3%A4ch-1002800/
Zuhören	https://pixabay.com/de/ohr-ohrmuschel-hören-zuhören-2973126/
“ärger”	https://pixabay.com/de/emoji-emotion-emoticon-ausdruck-1585197/
Anführungszeichen	https://pixabay.com/de/zitat-sprechblase-anf%C3%BChrungszeichen-1122421/
Fragezeichen	https://pixabay.com/de/fragezeichen-alphabet-frage-3246711/
“Freude”	https://pixabay.com/de/smiley-freude-umarmen-2674401/
Bilder Smileys:	https://commons.wikimedia.org/wiki/Smiley
Ton Lachen:	http://www.salamisound.de/3832006-langes-schaebiges-lachen

Ton Weinen:	http://www.salamisound.de/6923056-saeugling-baby-leichtes#
Hintergrund Pong:	https://pixabay.com/de/himmel-blau-wolken-berge-dunkel-34536/
Hintergrund Mole:	https://pixabay.com/de/gras-teppich-textur-struktur-gr%C3%BCn-1582945/
Maulwurf:	https://pixabay.com/de/maulwurf-m%C3%BCcke-einen-elefanten-tier-156762/
Eichhörnchen	https://pixabay.com/de/eichh%C3%B6rnchen-brown-cartoon-eichel-47528/

Bei den restlichen Abbildungen handelt es sich um Screenshots aus dem App Inventor <http://appinventor.mit.edu/>, welche teilweise bearbeitet wurden.

weiterführende Literatur:

Starting out with App Inventor for Android, Tony Gaddis and Rebecca Halsey. Pearson Education Limited 2015, ISBN-13: 978-1-292-08032-1

Android Apps mit Appinventor2: Jeder kann programmieren, Karl-Hermann Rollke, CreateSpace Independent Publishing Platform 2017, **ISBN-13:** 978-1544256009

Einführung in die Programmierung mittels App_Inventor

Bitte sparen Sie nicht mit Kritik!!!!

Dieser Fragebogen ist anonym und alle Angaben freiwillig.

Die erhobenen Daten sollen zur Bewertung und Verbesserung des Konzepts im Rahmen meiner Diplomarbeit dienen.

Bitte bewerten Sie nach dem Schulnotensystem

2. Bewertung der im Konzept vorhandenen Beispielaufgaben

Bitte bewerten Sie nach dem Schulnotensystem

[illegible]

3. Erfahrungen mit dem App_Inventor

Bitte bewerten Sie nach dem Schulnotensystem

	5	4	3	2	1	keine Angabe
Ich kannte den App_Inventor schon vorher	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Ich habe Erfahrungen mit dem Einsatz des App_Inventors im Unterricht	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

4. Einführung in die Programmierung in der Schule

Bitte bewerten Sie nach dem Schulnotensystem

	5	4	3	2	1	keine Angabe
Der App_Inventor ist für den Unterricht „Einführung in die Programmierung“ geeignet	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Mit diesem Konzept kann man die vom Lehrplan geforderten Kompetenzen vermitteln	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Ich erkenne den sozialen Mehrwert der mit diesem Konzept vermittelt wird	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

5. Warum finden Sie es wichtig bzw. nicht so wichtig Coding in der Schule zu unterrichten

Bitte um eine Antwort in eigenen Worten

6. Vermittlung sozialer Fähigkeiten mit Hilfe diese Konzepts

Bewerten Sie nach dem Schulnotensystem

	5	4	3	2	1	keine Angabe
Die Gruppengespräche sind eine sinnvolle Ergänzung	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Die Präsentationen sind ein wichtiger Bestandteil des Konzepts	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Die Vermittlung von Soft Skills ist im Informatikunterricht wichtig	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Die Thematisierung personenzentrierter Kommunikation und aktives Zuhörens ist ein wichtiger Bestandteil dieses Konzepts	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

7. Meine Rolle als Lehrperson

	5	4	3	2	1	keine Angabe
Ich sehe mich eher als Instruktor	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Ich sehe mich eher als Lernbegleiter	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Ich kann mir gut vorstellen in die Rolle des „Lernbegleiters“ zu schlüpfen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

8. Was ich dazu noch sagen möchte

9. Verwendbarkeit des Konzepts

Bitte bewerten Sie nach dem Schulnotensystem

	5	4	3	2	1	keine Angabe
Ich werde dieses Konzept vermutlich im Unterricht einsetzen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Ich werde dieses Konzept Kollegen und Kolleginnen empfehlen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

10. Verbesserungsvorschläge, Hinweise und Anregungen

Was ich zu diesem Konzept noch sagen möchte

11. Wie lange unterrichten Sie bereits

Bitte wählen Sie aus

- ☐ weniger als 5 Jahre
- ☐ 5 – 10 Jahre
- ☐ 10 – 15 Jahre
- ☐ mehr als 15 Jahre

☐ keine Angabe

12. Welche Fächer unterrichten Sie noch

Mehrfachauswahl ist möglich

Mein Zweitfach ist

13. An welchem Schultyp unterrichten Sie im Fach Informatik

Mehrfachangaben sind möglich

Bitte wählen Sie aus

- ☐ AHS
- ☐ BORG
- ☐ HTL
- ☐ HAK
- ☐ andere

☐ keine Angabe

Vielen Dank für Ihre Bereitschaft dieses Konzept zu begutachten und den Fragebogen auszufüllen!

Ich möchte mich ganz herzlich für Ihre Mithilfe bedanken.

Ihre Antworten wurden gespeichert, Sie können das Browser-Fenster nun schließen.