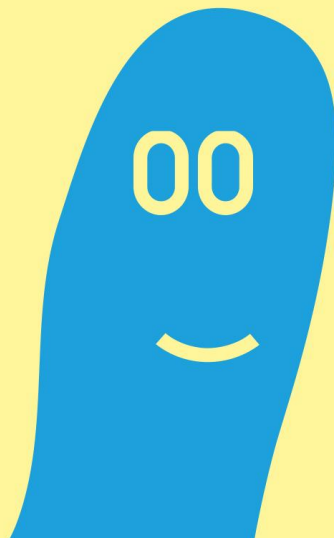




**CODING FOR
INCLUSION**

www.codinc.fun

 Co-funded by the
Erasmus+ Programme
of the European Union



Methodische Handreichung

EINLEITUNG	3
HINTERGRUND	4
WEITERENTWICKLUNG DES KONZEPTS	4
DIE SPEZIFISCHEN ZIELE VON CODINC:	4
CODINC RICHTET SICH DIREKT AN:	5
LERNZIELE	6
LERNZIELE FÜR LEHRKRÄFTE IN DER SEKUNDARSTUFE UND GRUNDSCHULE	6
LERNZIELE FÜR DIE JUGENDLICHEN	7
INFORMATISCHES DENKEN	7
KOOPERATIONS- UND SOZIALKOMPETENZ	8
KREATIVE FÄHIGKEITEN	9
PROBLEMLÖSUNGS- UND GESTALTUNGSFÄHIGKEITEN	9
PÄDAGOGISCHE ZIELE	9
LERNZIELE FÜR GRUNDSCHULKINDER	9
LERNZIELE IM BEREICH GRUNDSCHULE	10
LERNZIELE FÜR DIE SCHÜLERINNEN UND SCHÜLER	10
INFORMATISCHES DENKEN	10
KOOPERATIONS- UND SOZIALKOMPETENZ	11
KREATIVE FÄHIGKEITEN	12
PROBLEMLÖSUNGS- UND DESIGNFÄHIGKEITEN	12
LERNZIELÜBERSICHT FÜR DIE GRUNDSCHULE	12
HINTERGRUND	13
KOMPETENZEN DES 21. JAHRHUNDERTS	13
INFORMATISCHES DENKEN	14
WAS IST INFORMATISCHES DENKEN?	14
WARUM IST INFORMATISCHES DENKEN SO WICHTIG?	14
KREATIVES DENKEN	15
WARUM PROGRAMMIEREN LERNEN?	15
PÄDAGOGISCHE GRUNDPRINZIPIEN UND MODERATION	18
HINTERGRUND	21
ARBEITSBLÄTTER	23
LINKS	23
WERKZEUGE	23
ANHANG I: REFERENZEN	24

EINLEITUNG

Diese methodische Handreichung richtet sich an Lehrkräfte in Sekundar- und Grundschulen und an Trainerinnen und Trainer der in der medienpädagogischen Arbeit mit Kindern und Jugendlichen.

Das Heft zeigt methodische Vorgehensweisen auf, um Schülerinnen und Schülern aus Sekundarschulen (ab ca. 13 Jahren) dabei zu unterstützen, Grundschulkindern im Alter von 8 bis ca. 12 Jahren erste Schritte beim Erwerb von Programmierfähigkeiten sowie Fähigkeiten im Bereich der naturwissenschaftlichen, technischen, mathematischen und künstlerischen Bildung (engl. STEAM: Science, Technical, Engineering, Arts and Mathematics¹) zu vermitteln.

Ziel des Projekts CODINC ist es, Jugendliche und ihre Lehrkräfte in den Grundlagen des Programmierens und in pädagogischen Fähigkeiten auszubilden, dass sie Coding-Workshops für Kinder in Grundschulen durchführen können. Mit unseren Trainerinnen und Trainern durchlaufen sie dafür vorab ein gemeinsames 10 Stunden Training.

Nach dieser Ausbildung werden die Jugendlichen bei der Durchführung der Workshops in der Grundschule von ihren Lehrkräften und unseren Trainern begleitet.

Die Lehrkräfte in den Sekundarschulen stärken ihre eigenen Kenntnisse in dem Bereich Programmieren, und entwickeln ein pädagogisches Konzept um durch begleitendes Coaching ihre Schüler zu befähigen, mit Grundschulkindern zu programmieren.

Das CODINC-Projekt konzentriert sich mit seinem Peer-to-Peer Konzept auf die Förderung der sozialen Inklusion. Kinder und Jugendliche, die im Rahmen des CODINC-Projekts angesprochen werden, kommen insbesondere aus sozial benachteiligten Umgebungen. Das Projekt CODINC trägt dazu bei, das Selbstbewusstsein und Selbstvertrauen dieser Jugendlichen zu stärken, indem es ihnen dabei hilft, ihr kreatives Potenzial zu entdecken. Dies hilft Schülerinnen und Schülern bei der ihrer Entwicklung ihrer Persönlichkeit auf ihrem Weg, einen Platz in der Berufswelt zu suchen. Die Zielgruppe Jugendliche benötigt besonders das Vertrauen und die Zuversicht von Erwachsenen, um auf dem Weg in das Berufsleben Orientierung zu erhalten.

¹ https://en.wikipedia.org/wiki/STEAM_fields

Hintergrund

Das Projekt CODINC nutzt das Konzept und die Erfahrungen, die der Partner MAKS mit seinem Vorhaben "Capital Digital" in einem der als benachteiligt eingestuften Stadtteile in Brüssel, Belgien, bei der Entwicklung und Umsetzung gesammelt hat.

Capital Digital war ein Programm, das eine innovative Peer-to-Peer-Lernmethode angewendet hat, indem es junge Menschen aus sozial benachteiligten Gebieten im außerschulischen Bereich zu so genannten E-Facilitatoren, also zu Trainern für jüngere Schülerinnen und Schüler, ausbildete. In der Praxis hat „Capital Digital“ 15-18-jährige Jugendliche erfolgreich darin geschult, mit Kindern in Grundschulen Coding-Workshops durchzuführen. Die Jugendlichen E-Facilitatoren lernten auf spielerische Art und Weise, Kinder für STEAM und Programmieraktivitäten zu begeistern.

„Capital Digital“ war die erste „Arbeitserfahrung“ zur beruflichen Orientierung für diese Jugendlichen, die ihr Selbstbewusstsein enorm gestärkt hat. Das Programm ermöglichte es ihnen, konstruktiv miteinander zu arbeiten um in die Rolle des „Lehrers“ für die Grundschulkinder zu schlüpfen und Verantwortung für ihren Erfolg zu übernehmen. Sie erweiterten dabei ihren Horizont bei der Entscheidung für einen passenden Ausbildungsweg.

Darüber hinaus unterstützte das Projekt junge Menschen durch die Entwicklung ihrer Fähigkeiten zum kritischen Denken, ihrer Kreativität sowie ihrer Kooperationsfähigkeiten und ihrer Offenheit gegenüber Wissenschaft und (digitaler) Bildung. Die pädagogische Methode des „Capital Digital“ Peer-Learning hat ein starkes Inklusionspotenzial zur Förderung von benachteiligten Schülern innerhalb und außerhalb des Klassenzimmers entfaltet.

Weiterentwicklung des Konzepts

Das CODINC-Projekt integriert und transformiert die Methodik und das Trainings-Toolkit von „Capital Digital“ in einen anderen Bildungskontext für das formale Bildungssystem der Sekundar- und Grundschulen. In fünf europäischen Ländern (Belgien, Zypern, Deutschland Italien und Spanien) erproben die beteiligten Partner das erweiterte Konzept.

Die spezifischen Ziele von CODINC:

1. Unterstützung der Lehrkräfte an Sekundarschulen zur Förderung benachteiligter Jugendlicher im Bereich STEAM mit einem integrativen Bildungsansatz, der auf Peer-Learning basiert.
2. Unterstützung von Jugendlichen aus sozial benachteiligtem Umfeld beim Erwerb und der Entwicklung ihrer Digital Skills, Kooperationsfähigkeiten sowie Problemlösungsfähigkeiten. Stärkung ihres Selbstbewusstseins und ihrer Kreativität durch ein Peer-Learning-Trainingsprogramm zum Thema Programmieren für Kinder.
3. Förderung der Entwicklung einer europäischen Community "Coding for Inclusion" zwischen den verschiedenen Beteiligten und Sektoren (formale und non-formale Bildung), zur Verwertung der Projektergebnisse.

Europaweit besteht die Notwendigkeit, die digitalen Kompetenzen aller Bürger systematisch zu fördern. In vielen Ländern sind neue digitale Technologien noch nicht ausreichend in die Bildungssysteme integriert. Insbesondere in sozial benachteiligten Gebieten bieten digitale Instrumente zusätzliche Chancen für Integration und Bildung. Während sich nahezu alle Kinder und Jugendlichen aktiv im Internet und sozialen Netzwerken bewegen (meist mit Smartphones), müssen Lehrer insbesondere methodische Fähigkeiten entwickeln, um Schüler bei der Nutzung von neuen Technologien im Kontext des Lernens angemessen zu unterstützen. Programmieren im Kontext zu STEAM bietet neue Möglichkeiten, das Interesse von Kindern und Jugendlichen zu wecken und Unterricht innovativ zu gestalten. Ziel ist, Kindern und Jugendlichen frühzeitig Chancen zu eröffnen, selbst zu kreativen Produzenten und Mitgestaltern der digitalen Welt zu werden, statt lediglich Konsumenten von digitalen Inhalten zu sein.

CODINC richtet sich direkt an:

- Grundschüler (Kinder im Alter von acht bis 12 Jahren) und Sekundarschüler (Jugendliche im Alter 15 bis 18 Jahren) aus benachteiligten Gebieten.
- Lehrerinnen und Lehrer an Grund- und weiterführenden Schulen.
- Ausbilder, die mit Jugendlichen in formalen, nicht-formalen und informellen Umgebungen arbeiten (Schulen, Medienzentren, Jugendzentren, Ausbildungszentren usw.).

Hauptbeteiligte sind Schulen, darüber hinaus richtet sich das Vorhaben auch an Familien, Bildungseinrichtungen und Bildungsanbieter, Behörden, politische Entscheidungsträger Organisationen der Zivilgesellschaft, NGO's, usw.

Diese methodische Handreichung stellt einen Trainingsplan für die Durchführung eines 10-stündigen Trainingskurses für die Jugendlichen und ein Curriculum für die Schüler und Lehrer bereit. Dabei gibt es viele Gemeinsamkeiten zwischen dem Programm für die Lehrkräfte und dem für die Jugendlichen, weil die Schülerinnen und Schüler nach Möglichkeit die gleichen Module für das Training nutzen sollten, die die Lehrkräfte ihrerseits für den Unterricht verwenden. Das Ausbildungsprogramm für Jugendlichen und ihre Lehrer konzentriert sich insbesondere auf die pädagogisch-didaktischen Aspekte und auf Methoden für das Programmieren lernen. Darüber hinaus vermittelt es Motivationstechniken und Anregungen zur Überprüfung der Wirksamkeit (Evaluation). Gleichzeitig hilft diese Vorbereitung dabei, ihnen die Sorge zu nehmen, dass sie in den Workshops mit den Kindern etwas trainieren müssen, was sie selbst nicht richtig verstehen. Mithilfe der leicht verständlichen Konzepte und durch das intensive Training werden die Schülerinnen und Schüler in ihrem Selbstvertrauen bestärkt. Es trägt dazu bei, sie von ihren eigenen Fähigkeiten zu überzeugen sodass ihr Selbstwertgefühl während des Projekts wachsen kann.

Viel Spaß beim Lesen und beim Ausprobieren!

LERNZIELE

Diese methodische Handreichung unterstützt Sie und Ihre Schülerinnen und Schüler dabei, das Spektrum an Aufgaben und Herausforderungen zu bewältigen um die Workshops mit Grundschulkindern durchzuführen: von der Entwicklung des so genannten "Informatischen Denkens", (gemeint ist nicht nur logisches Denken, sondern ein "digitales Verständnis" zu entwickeln) bis hin zur Ausprägung von kreativen und sozialen Fähigkeiten.

Der methodische Leitfaden ist in zwei Abschnitte unterteilt, einen für die Primar- und einen für die Sekundarstufe. Jeder Abschnitt enthält Lernziele für die am Programm teilnehmenden Grundschulkindern oder die Jugendlichen und separate Ziele für Sie als Lehrkraft.

Nutzen Sie die Lernzielübersichten, um einen schnellen Überblick darüber zu erhalten, welche Aktivitäten zur Vervollständigung der einzelnen Lernziele beitragen.

Lernziele für Lehrkräfte in der Sekundarstufe und Grundschule

Als Lehrkraft erhalten Sie Einblicke, Anregungen und erweitern Ihre Kompetenzen zum so genannten Informatischen Denken, zum Programmieren sowie zur Pädagogik.

Sie entwickeln ein besseres Verständnis davon,

- was Informatisches Denken ist und welche Bedeutung es für die gesellschaftliche Entwicklung hat.
- welche wichtigen Schlüsselkompetenzen (21st Century Skills) sind und welche Bedeutung sie für die Zukunftsfähigkeit der Kinder und Jugendliche haben?
- wie Programmieren die Fähigkeit zum Informatischen Denken, die Problemlösungskompetenz aber auch die Kreativität von Kindern und Jugendlichen fördern kann?
- welche essentiellen pädagogischen und sozialen Fähigkeiten Jugendliche entwickeln sollten, um Workshops für Grundschulkindern zum Programmieren lernen zu leiten.

Als Pädagogin/Pädagoge werden Sie in folgenden Bereichen aktiv arbeiten:

- Nicht digitales (offline-) Programmieren
- Programmieren mit Hilfe von kindgerechten (digitalen) Programmierertools
- Übungen und Aktivierungen zur Kooperationsfähigkeit, zum kollaborativen Arbeiten und zur Kreativität
- Gestaltung von Lernangeboten
- Pädagogische Vorgehen zur Motivation und Aktivierung.

Die Lehrkräfte der Sekundarschule trainieren gemeinsam mit ihren Schülern; die Lehrkräfte der Grundschule absolvieren ein separates Training zur Qualifizierung.

Lernziele für die Jugendlichen

Die am Programm teilnehmenden Schülerinnen und Schüler erwerben Fähigkeiten in den Bereichen:

- logisches Denken / Informatisches Denken,
- Team- und kooperationsfähigkeit sowie Sozialkompetenz,
- kreative Fähigkeiten,
- Problemlösungs- und Gestaltungsfähigkeiten,
- pädagogische Fähigkeiten und erste Einblicke in Gruppendynamik.

Mit dieser Handreichung erlangen Ihre Schülerinnen und Schüler auch Basis-Fähigkeiten, einfache Programmiersprachen und visuelle Oberflächen zum Programmieren auf kreative Weise zu verwenden. Sie wenden Mathematik an und können sich entscheiden, das Abschlussprojekt mit einem Schulfach Ihrer Wahl, wie z. B. Geschichte, Geographie, Kunst etc. zu verknüpfen.

Informatisches Denken

Die Lernziele für das Informatische Denken gliedern sich in zwei Abschnitte:

- die für die Jugendlichen nachzuvollziehende konzeptionelle Ebene und
- deren Verknüpfung mit der kognitiven Ebene (praktische Umsetzung).

Diese Handreichung bildet neun **Schlüsselkonzepte** zum Informatischen Denken ab, mit denen Jugendliche erfolgreich Programmieren lernen und erste eigene Spiele oder Programme erstellen können.

Konzept	Beschreibung
Algorithmen	Eine Strategie des Vorgehens, ähnlich wie bei einem Kochrezept, z. B. für ein Computerprogramm, für ein Spiel, eine App usw.
Sequenzen	Befehle, die in direkter Reihenfolge hintereinander gegeben werden. Ein Computer führt sie der Reihe nach aus.
Wiederholung und Schleifen	Eine Teilbereich von Anweisungen, die mehrmals (oder unendlich) wiederholt werden, wird als Schleife bezeichnet.
Events und Auswahl	Eine Inidkation, wann ein Ereignis stattfinden soll; z.B. soll in einem Spiel die Katze beginnen, sich zu bewegen, wenn der"Start"-Button gedrückt wird.
Bedingungen und logische Operatoren	Der Computer kann nach dem „Wenn-dann“-Prinzip eine Entscheidung treffen. Wenn etwas passiert, dann sollte ein Ereignis stattfinden, oder nicht.
Mathematische Operatoren	Algorithmen, die Rechenvorgänge (z. B. Multiplikation oder Addition) erfordern, beispielsweise um in einem Spiel die Geschwindigkeit einer Kugel über einen festgelegten Zeitraum zu reduzieren.

Variablen- und Datenverwaltung	Variablen sind Felder, in die Zahlen (oder Texte) eingegeben werden können. Diese Felder werden dann im Code verwendet, um z. B. die aktuelle Punktzahl des Spielers zu speichern.
Funktionen	Wiederverwendung eines Teilbereichs eines Codes, z. B. zum Vorwärtsgen,

Folglich gibt es sechs entscheidende **Denkweisen**, die zum Informatischen Denken beitragen und dabei helfen, ein kompetenter Programmierer zu werden. Diese Denkweisen sind nicht nur im Bereich des Programmierens hilfreich, sondern lassen sich auch gut auf andere Bereiche des Denkens und Lebens, wie z.B. die kreative Problemlösung, übertragen.

Denkweisen	Beschreibung
Inkrementelle und iterative Arbeitsstrategien	Die Schülerinnen und Schüler teilen ihre Arbeit in kleine Schritte auf und kehren zu den vorhergehenden Schritten zurück, um ihr Programm zu verbessern. Programmieren ist kein linearer Prozess, sondern verläuft wie in einer Spirale mit Wiederholungen.
Testen und Debuggen	Die Schülerinnen und Schüler können ein Spiel oder Projekt hinsichtlich ihrer Erwartungen testen und daraus Rückschlüsse für Verbesserungen ziehen. Sie können Probleme (so genannte Bugs) identifizieren und mit ihren eigenen Strategien beheben („Debugging“).
Wiederverwendung und Neuzusammensetzung	Die Schülerinnen und Schüler lernen von Projekten anderer, sie können Teile des Codes oder Gedankenprozesse daraus selbst wiederverwenden (remix) und dabei bestehende Lösungen oder Projekte in etwas Neues umwandeln.
Abstraktion	Die Schüler können die aus einem Projekt gewonnenen Erkenntnisse in abstrakte Muster umsetzen.
Modularisierung	Die Schüler lernen, Projekte in kleinere Teile zu zerlegen (z.B. Bewegung, Geschwindigkeit, Partitur...) und diese Teile in ihrer Arbeit wiederzuverwenden.
Informationssammlung und Management	Schüler können Informationsquellen identifizieren und an verschiedenen Orten nach Lösungen für ihre Probleme suchen, z. B. mit Gleichaltrigen oder online.

Kooperations- und Sozialkompetenz

Die methodische Herangehensweise im CODINC-Projekt basiert auf einem kollaborativen Peer-to-Peer-Ansatz, bei dem ältere Schülerinnen und Schüler (Sekundarstufe) zusammenarbeiten, um jüngere Schülerinnen und Schüler (Grundschule) zu unterrichten.

Ihre Schüler werden ihren sozialen Fähigkeiten ausbauen, weil sie als Team arbeiten und gemeinsam lernen. Sie werden lernen, miteinander und mit den Grundschulern, die sie trainieren, zu verhandeln. Ihre Unterrichtstätigkeit erfordert nicht nur Fachkenntnisse zum Programmieren, sondern ermöglicht auch, über pädagogische Fragen nachzudenken und Einblicke in das Denken junger Menschen zu erhalten. Außerdem lernen sie, ihre eigenen

Denkprozesse in einer Sprache zu beschreiben, die für ihre Zielgruppe geeignet ist (sei es die Gruppe der jüngeren Schüler oder ihrer Altersgenossen). Schließlich lernen sie voneinander und von der jüngeren Gruppe, durch gegenseitigen Erfahrungsaustausch und gemeinsame Problemlösung.

Kreative Fähigkeiten

Während des gesamten Projekts werden die kreativen Fähigkeiten der Schülerinnen und Schüler aktiv gefördert. Die drei Hauptaspekte der Kreativität²:

- Ideen und Projekte anderer Menschen erforschen
- eigene Fähigkeiten, Ideen, Projekte etc. zu neuen kreativen Aktivitäten entwickeln
- vorhandene Objekte in etwas Neues verwandeln.

Problemlösungs- und Gestaltungsfähigkeiten

Die Jugendlichen lernen, wie man ein neues Problem und seine Komponenten identifiziert. Sie erwerben auch Strategien, um durch Brainstorming neue Ideen zu generieren und diese umzusetzen. Sie werden ihre Lösung begutachten und über die Wirkung nachdenken. Anschließend werden sie ihre Lösung optimieren und das Problem in einer nächsten Iteration des Prozesses neu bewerten.

Pädagogische Ziele

In dem sie zehnstündigen Trainingsprogramm erwerben die Jugendlichen verschiedene pädagogischen Fähigkeiten und Einblicke in die Gruppendynamik. Mehr noch lernen sie aber durch die praktische Arbeit mit einer Gruppe von Kindern.

Sie erwerben Kenntnisse und Fähigkeiten in Bezug auf die Gruppenanimation; die Steuerung der individuellen Motivation und der Gruppenmotivation bei und den Einsatz von „Energizern“ zur Steigerung der Stimmung; sowie Kenntnisse zur Bewertung von Aktivitäten, Prozessen und ihrer eigenen Leistung. Darüber hinaus lernen sie, wie Gruppenregeln gemeinsam festgelegt und eingehalten werden und nicht zuletzt organisatorische Fähigkeiten.

Lernziele für Grundschul Kinder

Die verschiedenen Aktivitäten und Methoden im CODINC-Projekt sind jeweils auf eine Auswahl dieser Lernziele bezogen. Mit der Durchführung des gesamten Programms werden alle Lernziele erreicht. Eine Übersicht über die verschiedenen Lernziele - und welche Aktivitäten sie erwerben werden - finden Sie in der **Übersicht Lernziele für die Sekundarstufe**

²<https://www.lego.com/r/legofoundation/-/media/lego%20foundation/downloads/foundation%20research/systematic%20creativity%20report.pdf?l.r2=-2137924029>

Lernziele im Bereich Grundschule

Lernziele für die Schülerinnen und Schüler

Die am Programm teilnehmenden Grundschul Kinder erwerben Fähigkeiten in den Bereichen:

- logisches Denken / Informatisches Denken,
- Team- und kooperationsfähigkeit sowie Sozialkompetenz,
- kreative Fähigkeiten,
- Problemlösungs- und Gestaltungsfähigkeiten,
- pädagogische Fähigkeiten und erste Einblicke in Gruppendynamik.

Mit dieser Handreichung erlangen Ihre Schülerinnen und Schüler auch Basis-Fähigkeiten, einfache Programmiersprachen und visuelle Oberflächen zum Programmieren auf kreative Weise zu verwenden. Sie wenden Mathematik an und können sich entscheiden, das Abschlussprojekt mit einem Schulfach Ihrer Wahl, wie z. B. Kunst- oder Sachkunde zu verknüpfen.

Informatisches Denken

Die Lernziele für das Informatische Denken gliedern sich in zwei Abschnitte:

- die für die Jugendlichen nachzuvollziehende konzeptionelle Ebene und
- deren Verknüpfung mit den kognitiven Ebene (praktische Umsetzung).

Diese Handreichung bildet neun **Schlüsselkonzepte** zum Informatischen Denken ab, mit denen Jugendliche erfolgreich Programmieren lernen und erste eigene Spiele oder Programme erstellen können.

Konzept	Beschreibung
Algorithmen	Eine Strategie des Vorgehens, ähnlich wie bei einem Kochrezept, z. B. für ein Computerprogramm, für ein Spiel, eine App usw.
Sequenzen	Befehle, die in direkter Reihenfolge hintereinander gegeben werden. Ein Computer führt sie der Reihe nach aus.
Wiederholung und Schleifen	Eine Teilbereich von Anweisungen, die mehrmals (oder unendlich) wiederholt werden, wird als Schleife bezeichnet.
Events und Auswahl	Eine Inidkation, wann ein Ereignis stattfinden soll; z.B. soll in einem Spiel die Katze beginnen, sich zu bewegen, wenn der"Start"-Button gedrückt wird.
Bedingungen und logische Operatoren	Der Computer kann nach dem „Wenn-dann“-Prinzip eine Entscheidung treffen. Wenn etwas passiert, dann sollte ein Ereignis stattfinden, oder nicht.

Mathematische Operatoren	Algorithmen, die Rechengänge (z. B. Multiplikation oder Addition) erfordern, beispielsweise um in einem Spiel die Geschwindigkeit einer Kugel über einen festgelegten Zeitraum zu reduzieren.
Variablen- und Datenverwaltung	Variablen sind Felder, in die Zahlen (oder Texte) eingegeben werden können. Diese Felder werden dann im Code verwendet, um z. B. die aktuelle Punktzahl des Spielers zu speichern.
Funktionen	Wiederverwendung eines Teilbereichs eines Codes, z. B. zum Vorwärtsgen

Folglich gibt es sechs entscheidende **Denkweisen**, die zum Informatischen Denken beitragen und dabei helfen, ein kompetenter Programmierer zu werden. Diese Denkweisen sind nicht nur im Bereich des Programmierens hilfreich, sondern lassen sich auch gut auf andere Bereiche des Denkens und Lebens, wie z.B. die kreative Problemlösung, übertragen.

Denkweisen	Beschreibung
Inkrementelle und iterative Arbeitsstrategien	Die Schülerinnen und Schüler teilen ihre Arbeit in kleine Schritte auf und kehren zu den vorhergehenden Schritten zurück, um ihr Programm zu verbessern. Programmieren ist kein linearer Prozess, sondern verläuft wie in einer Spirale mit Wiederholungen.
Testen und Debuggen	Die Schülerinnen und Schüler können ein Spiel oder Projekt hinsichtlich ihrer Erwartungen testen und daraus Rückschlüsse für Verbesserungen ziehen. Sie können Probleme (so genannte Bugs) identifizieren und mit ihren eigenen Strategien beheben („Debugging“).
Wiederverwendung und Neuzusammensetzung	Die Schülerinnen und Schüler lernen von Projekten anderer, sie können Teile des Codes oder Gedankenprozesse daraus selbst wiederverwenden (remix) und dabei bestehende Lösungen oder Projekte in etwas Neues umwandeln.
Abstraktion	Die Schüler können die aus einem Projekt gewonnenen Erkenntnisse in abstrakte Muster umsetzen.
Modularisierung	Die Schüler lernen, Projekte in kleinere Teile zu zerlegen (z.B. Bewegung, Geschwindigkeit, Partitur...) und diese Teile in ihrer Arbeit wiederzuverwenden.
Informationssammlung und Management	Schüler können Informationsquellen identifizieren und an verschiedenen Orten nach Lösungen für ihre Probleme suchen, z. B. mit Gleichaltrigen oder online.

Kooperations- und Sozialkompetenz

Die methodische Herangehensweise im CODINC-Projekt basiert auf einem kollaborativen Peer-to-Peer-Ansatz, bei dem ältere Schülerinnen und Schüler (Sekundarstufe) zusammenarbeiten, um jüngere Schülerinnen und Schüler (Grundschule) zu unterrichten.

Die Kinder lernen im Abschlussprojekt gemeinsam an einem Spiel zu arbeiten, miteinander zu verhandeln und ihre eigenen Denkprozesse zu beschreiben.

Außerdem lernen sie, ihre eigenen Projekte vor ihren Mitschülern zu präsentieren und sich gegenseitiges Feedback zu geben und Feedback aufzunehmen. Schlussendlich lernen sie nicht nur allein, sondern vor allem aus gemeinsamen Erfahrungen.

Kreative Fähigkeiten

Während des gesamten Projekts werden die kreativen Fähigkeiten der Schülerinnen und Schüler aktiv gefördert. Die drei Hauptaspekte der Kreativität³:

- Ideen und Projekte anderer Menschen erforschen
- eigene Fähigkeiten, Ideen, Projekte etc. zu neuen kreativen Aktivitäten entwickeln
- vorhandene Objekte in etwas Neues verwandeln.

Problemlösungs- und Designfähigkeiten

Schließlich befasst sich die Methodik auch mit Problemlösungs- und Designfähigkeiten.

Die Grundschul Kinder lernen, wie man ein neues Problem und seine Komponenten erkennt. Sie erarbeiten Strategien, um mittels Brainstorming neue Ideen zu entwickeln und wie sie umzusetzen können. Sie werden ihre Lösungen begutachten und über die Wirkung nachdenken. Anschließend werden sie ihre Lösung optimieren und das Problem in einer nächsten Iteration des Prozesses neu bewerten.

Lernzielübersicht für die Grundschule

Die verschiedenen Aktivitäten und Methoden im CODINC-Projekt sind jeweils auf eine Auswahl dieser Lernziele bezogen. Mit der Durchführung des gesamten Programms werden alle Lernziele erreicht. Eine Übersicht über die verschiedenen Lernziele - und welche Aktivitäten sie erwerben werden - finden Sie in der Matrix für die Sekundarstufe.

Vgl. Anhang Lernziele

³<https://www.lego.com/r/legofoundation/-/media/lego%20foundation/downloads/foundation%20research/systematic%20creativity%20report.pdf?l.r2=-2137924029>

HINTERGRUND

Kompetenzen des 21. Jahrhunderts

Das Weltwirtschaftsforum hat eine Untersuchung über die wichtigsten Kompetenzen im Jahr 2020 im Vergleich zu 2015 durchgeführt.

Kreativität wird zu einer der drei wichtigsten Kompetenzen, die Arbeitskräfte benötigen. Mit der Flut neuer Produkte, neuer Technologien und neuer Arbeitsweisen müssen die Arbeitnehmer kreativer werden, um von diesen Veränderungen zu profitieren.

Das aktive Zuhören, welches heute zu den Kernkompetenzen zählt, wird komplett aus den Top 10 verschwinden. Emotionale Intelligenz, die heute nicht in den Top 10 zu finden ist, wird zu einer der Top-Fähigkeiten, die alle brauchen.

[\(https://www.weforum.org/agenda/2016/01/the-10-skills-you-need-to-thrive-in-the-fourth-industrial-revolution/\)](https://www.weforum.org/agenda/2016/01/the-10-skills-you-need-to-thrive-in-the-fourth-industrial-revolution/)

Das Weltwirtschaftsforum erwähnte auch, dass 65 % der Kinder, die heute in die Schule gehen, in einem Beruf arbeiten werden, den es heute noch gar nicht gibt. Daraus folgt, dass Lernen eine ganz andere Dimension erhalten wird.

In Europa hat die OECD eine Reihe von Fähigkeiten gesammelt, die Kinder, Jugendliche und Erwachsene benötigen werden, um in der modernen Gesellschaft erfolgreich zu sein: die Fähigkeiten des 21. Jahrhunderts. Dabei handelt es sich um miteinander verknüpfte Kompetenzen wie Problemlösung, Zusammenarbeit, kreatives Denken und Umgang mit digitalen Medien. Diese sollten Teil jeder Bildung sein und die Bürger der Zukunft unterstützen, um sie für das lebenslange Lernen vorzubereiten.

Diese Fähigkeiten sind:

1. Beurteilungsvermögen
2. Schöpferische Fähigkeit
3. Problemlösungsfähigkeit
4. Informatisches Denken
5. Beurteilungsvermögen
6. Basis IT-Kenntnisse
7. Umgang mit digitalen Medien
8. Kommunikationsfähigkeit
9. Kooperationsfähigkeit
10. Entscheidungsfähigkeit und Eigenständigkeit
11. Selbstreflexion

In der CODINC-Methodik zielen wir ausdrücklich darauf ab, diese Fähigkeiten zu stärken und zu fördern. Dabei konzentrieren wir uns insbesondere auf informatisches Denken, Problemlösungsfähigkeit, Kreativität, Zusammenarbeit und Kommunikation. Auch die anderen Kompetenzen werden im Projekt berücksichtigt..

Informatisches Denken

Was ist Informatisches Denken?

Wenn man den Begriff "Informatisches Denken" sieht, könnte man meinen, er sei ein Teil der Informatik. Er ist jedoch noch viel weitläufiger gefasst. Der Begriff wurde 2006 von Jeanette Wing an der Carnegie Mellon University geprägt.

Sie können im Internet viele Erläuterungen zu diesem Thema finden, wir empfehlen Ihnen diese kurze Erklärung: <https://www.youtube.com/watch?v=eJ1GskpiLKc>

Informatisches Denken ist nicht nur eine Kompetenz, sondern beinhaltet eine Reihe von Konzepten, Anwendungen, Werkzeugen und Denkstrategien, die zur Lösung von Problemen eingesetzt werden. Diese können verwendet werden, um das menschliche DNA-Genom zu entdecken oder um die Schriften von Shakespeare zu analysieren. Sie können Informatisches Denken üben, ohne jemals einen Computer zu benutzen. Aber was ist es dann?

"Informatisches Denken ist ein Ansatz zur Problemlösung. Es nimmt ein Problem auseinander und findet heraus, wie man es lösen kann, indem es das benutzt, was wir über Berechnung wissen." (Diane Main, Direktorin für Didaktik, Innovation und Design, The Harker School, Oberstufe)

Hauptaspekte des informatischen Denkens:

1. Zersetzung: Zerlegung eines Problems in kleinere Teile;
2. Mustererkennung: Ähnlichkeiten und Unterschiede zwischen den verschiedenen Teilen finden, um Vorhersagen treffen zu können;
3. Abstraktion: die Fähigkeit, die allgemeinen Prinzipien hinter den Teilen und Mustern in Problemen zu finden;
4. Algorithmus-Gestaltung: Entwicklung der Schritt-für-Schritt-Anleitung zur Lösung verschiedener Probleme.

Warum ist Informatisches Denken so wichtig?

In einer Gesellschaft, die immer komplexer und technologieorientierter wird, ist es für Kinder und Jugendliche notwendig zu lernen, kritisch zu denken und in der Lage zu sein, ihre eigenen digitalen Erfahrungen zu kontrollieren und zu gestalten. Wir wollen, dass Kinder nicht die Konsumenten der digitalen Technologie sind, sondern dass sie die Produzenten werden und ihnen ein kritisches Verständnis beibringen. Informatisches Denken wird dann Teil einer größeren Medienkompetenz.

Die derzeitige digitale Kluft wird sich nicht nur vergrößern, wenn wir diese Notwendigkeit ignorieren, sondern sie wird sich von der Nutzung der Technologie hin zur Schaffung von Technologie verlagern. Die "Profiteure und nicht-Profiteure" werden zu den "Kreativen und Schöpferischen und nicht-Schöpferischen". Es gäbe eine Teilung zwischen denen, die ihren eigenen Technologieeinsatz kontrollieren und Lösungen für ihre Bedürfnisse generieren können, und denen, die einfach ohne kritisches Nachdenken konsumieren.

Eine Hoffnung liegt darin, dass der frühe Kontakt mit den Herausforderungen des informatischen Denkens die Kinder dazu inspirieren könnte, sich für weitere Studiengänge im Bereich STEAM (Science-Technology-Engineering-Arts-Mathematik) zu entscheiden, insbesondere für Mädchen, die ein positives Feedback und ein gesteigertes Selbstwertgefühl in diesen Bereichen erfahren.

Natürlich gibt es auch den beruflichen Bedarf. Der Arbeitsmarkt wird mehr und mehr Mitarbeiter benötigen, die in den Bereichen IKT und Kodierung ausgebildet sind. Schon jetzt gibt es einen Mangel. Aber nicht nur Programmier- oder Technik-Jobs brauchen rechnerisches Denken. Mit der Übernahme "einfacher" Aufgaben durch Künstliche Intelligenz und Robotik werden immer mehr Mitarbeiter mit kreativen und problemlösenden Fähigkeiten benötigt.

Kreatives Denken

Was ist Kreativität? Im Essay "Systematische Kreativität im digitalen Raum" (Ackermann et al., 2009) wird das kreative Spiel in drei Hauptaktivitäten unterteilt: Exploration, Kombination und Transformation.

Kombinieren - mit neuen, überraschenden und wertvollen Ideen und Artefakten durch bestehende Ideen und Objekte zu kombinieren.

Entdecken - Erweiterung unseres Verständnisses von einem Bereich oder kreativer Bereiche durch die Integration überraschender und wertvoller Ideen und Artefakte.

Transformieren - die Art und Weise, wie wir die Welt sehen oder verstehen, mit neuen, überraschenden und wertvollen Ideen und Artefakten.

Innerhalb der 21st Century Skills wird die Fähigkeit des kreativen Denkens in mehrere kleinere Fertigkeiten unterteilt, wie das Kennen und Anwenden kreativer Techniken; das Verlassen der ausgetretenen Pfade; das Sehen neuer Zusammenhänge und Kombinationen; das Wagnis, (kalkulierte) Risiken einzugehen; das Sehen von Fehlern als Lernmöglichkeiten und eine offene, nachforschende Haltung.

Warum Programmieren lernen?

Es gibt viele Möglichkeiten, die Fähigkeiten des 21. Jahrhunderts zu entwickeln, insbesondere informatisches Denken und Kreativität. Wir glauben jedoch, dass es eine der besten Wege ist,

Kindern und Jugendlichen das Codieren beizubringen. Das liegt daran, dass es ihre Fähigkeit stimuliert, Muster zu erkennen, zu entwickeln und zu erstellen. Kinder, die lernen zu programmieren, haben ein besseres Verständnis für Systeme und deren Gestaltung. Sie analysieren ständig Probleme und entwickeln neue Lösungen für sie. Das Codieren bietet auch Kindern die Möglichkeit, selbstständig mit der Erstellung zu beginnen.

Das Verständnis von Code und den Prinzipien der Programmiersprachen ist ein wesentlicher Vorteil für zukünftige Arbeitgeber und Arbeitssuchende. Das bedeutet aber nicht, dass sich der Vorteil der Kodierung auf berufliche Zwecke beschränkt.

Das Erlernen von Code ermöglicht es Kindern, Schöpfer und Produzenten in einer digitalen Welt zu werden. Statt täglich mit unkritischer Einstellung Technologie zu konsumieren, können sie nun ihre eigene digitale Zukunft aufbauen. Statt App-Konsumenten werden sie zu App-Entwicklern. Ob beruflich oder einfach nur zum Spaß, das Erlernen von Code ist ein großer Vorteil.

Warum ist das Programmieren mit benachteiligten Kindern und Jugendlichen wichtig?

Reicht der Zugang zu Computern aus, um die digitale Kluft zwischen höheren sozialen Schichten und benachteiligten Gruppen zu verringern? Viele Berichte heben die Notwendigkeit hervor, den Umgang mit dem Computer oder anderen Geräten zu erlernen.

Der Zugang zu Informationen und der Besitz der entsprechenden Informationskompetenz sind zu einem Schlüsselfaktor im wirtschaftlichen und sozialen Leben geworden und können zu einer neuen Quelle der Schichtung in der Gesellschaft führen. Die "Wissenslücke" ist kein neues Konzept (Tichenor, Donohue und Olien 1970), kann aber im digitalen Zeitalter an Bedeutung gewinnen (Bonfadelli 2002). Die Verbreitung der digitalen Technologie und der IKT hat in der Geschichte des menschlichen Lebens beispiellose Ausmaße angenommen. Informationen werden im Internet und auf digitalen Geräten immer zugänglicher. In den meisten Studien nimmt die so genannte erste digitale Kluft (d.h. die sozioökonomische Kluft beim Zugang zur Technologie) ab und wird mit der Reifung des Digitalisierungsprozesses als verschwunden angesehen. viel Technik im häuslichen Umfeld führt nicht zu einer besseren Nutzung und höheren Qualifikation. Höhere soziale Schichten können ihre höheren Ressourcen und ihr kulturelles Kapital einsetzen, um mehr von den neuen Technologien zu profitieren und so einen schnelleren und besseren Zugang zu Informationen zu erreichen.

Studenten höherer Gesellschaftsschichten nutzen Technologien besser aus als ihre benachteiligten Altersgenossen, die den Computer zumeist als Verbraucher nutzen, um Spiele zu spielen, Facebook zu nutzen und Produkte über das Internet zu kaufen.

Der Ausbau der digitalen Kompetenzen benachteiligter Gruppen ist äußerst wichtig, um diese digitale Kluft zu verringern und die soziale Integration und Mobilität zu fördern. Die Produktion von Multimedia-Produkten mit mobilen Geräten oder Computern verändert die Beziehung

zwischen dem Lernenden und den Geräten. Wenn man IKT-Inhalte produziert, wird man zum Prosumenten und lernt, den Computer als Werkzeug für lebenslanges Lernen zu betrachten. Das ist auch eines der indirekten Ziele dieses Projekts.

Programmieren mit Jugendlichen: Steigerung des Selbstwertgefühls und des Selbstvertrauens.

Benachteiligten Jugendlichen eine Rolle im schulischen Umfeld zu geben, bedeutet ihnen eine Stimme und ein Gefühl der Zugehörigkeit zur Schulgemeinschaft zu geben. Die Peer-to-Peer-Methodik ist eine geeignete Wahl, um ihr Wohlbefinden zu steigern und jüngere Menschen zu stärken. Jugendlichen mit Migrationshintergrund und Flüchtlingshintergrund darf nicht das gleiche wissenschaftliche oder soziale Kapital wie ihren Altersgenossen angeboten werden, wodurch die Gefahr höherer Abbrecherquoten und eines geringeren Selbstwertgefühls besteht. Das CODINC-Projekt bietet Jugendlichen mit benachteiligten Hintergründen wie Flüchtlingen oder Drittländern wissenschaftliches und soziales Kapital, wenn sie für die Rolle eines Kodierungs-/STEAM-Moderators ausgebildet werden. Dies ermöglicht es ihnen, ihr Selbstvertrauen zu stärken und sich selbst als Akteure in der digitalen Welt und Führer in ihrer Gemeinschaft zu sehen. Die CODINC-Erfahrung kann auch die Lehrer-Schüler-Beziehung im Klassenzimmer verbessern, da die Lehrer die Möglichkeit haben, die Schüler auf eine andere Art und Weise in die Ausbildung einzubinden und eine andere Rolle zu erleben. Dies kann auch dazu beitragen, das Niveau der Zusammenarbeit und des sozialen Zusammenhalts im Unterricht zu erhöhen.

Pädagogische Grundprinzipien und Moderation

In der CODINC-Methodik basieren die pädagogischen Kernprinzipien auf den vier von Mitch Resnick (MIT, LifeLong Kindergarten und Schöpfer der visuellen Programmiersprache Scratch) übermittelten P's: Projekte, Leidenschaft, Peers und Spiel.

Sein kurzes Einführungsvideo zu Creative Learning und den vier P's finden Sie hier: <http://bit.ly/resnickfourp>

Projekte

Kinder und Jugendliche sind am meisten motiviert (und lernen am besten), wenn sie an Projekten arbeiten, die sie lieben und die für sie von Bedeutung sind. Daher ist das Programm nicht auf Tutorials oder folgende Beispiele beschränkt, sondern führt zum Aufbau eines persönlichen Projekts. Kinder kreieren ihr eigenes Spiel, das sie selbst entwickelt haben und auf das sie stolz sein können.

Leidenschaft

Lernen macht Spaß, oder sollte es zumindest sein. Im 10-Stunden-Programm ist es eines der Hauptziele, dass sowohl die Kinder als auch die Jugendlichen Spaß haben und das, was sie tun, lieben. Die Leidenschaft für ein Thema oder Projekt führt zu mehr Motivation, mehr Engagement und tieferem Lernen.

Gleichaltrige

Es ist kein Zufall, dass das Programm einen Peer-to-Peer-Ansatz für das Lernen anwendet. Zusammenarbeiten und spielen ist einer der stärksten Lernprozesse, die es gibt. Die Schülerinnen und Schüler der Sekundarstufe werden nicht nur ihre jüngeren Altersgenossen unterrichten, sondern auch von ihnen lernen. Solitäres Lernen ist nicht nur langsamer und restriktiver, sondern auch langweilig.

Spielen

Dieses Programm besteht nicht aus traditionellem Unterricht. Ziel ist es, mit Technik, mit Konzepten und miteinander zu spielen. Durch aktives Experimentieren und lustige Spiele lernen die Kinder mehr als beim Sitzen und Zuhören. Deshalb sollte die ideale Atmosphäre spielerisch und lustig sein, nicht restriktiv oder traditionell.

Programmieren und die Makerbewegung basieren auf den pädagogischen Prinzipien von Pädagogen wie Jean Piaget und Paolo Freire und gehen von folgenden Gesichtspunkten aus:

Kinder sind natürliche Lernende

Ihr Wunsch, die Welt um sie herum zu verstehen, Fähigkeiten und Kompetenzen in dieser Welt zu erwerben und eine sinnvolle Rolle darin zu spielen, ist so stark wie ihr Wunsch nach Nahrung, Wärme, Komfort und Liebe.

Wie machen sie das? Sie lernen durch das Eintauchen in eine menschliche Kultur und durch genaues Beobachten, Spielen, Nachahmen und Mitmachen in dieser Kultur. Sie sehen, was ältere, erfahrenere Menschen tun, und sie haben einen starken Drang, die gleichen Dinge zu tun. Tatsächlich explodieren kleine Kinder vor Wut, wenn es ihnen nicht erlaubt ist, Dinge zu tun, die sie andere tun sehen. Sie wollen an dem Tanz "Leben" teilnehmen, nicht nur an der Seitenlinie. Kinder sind wie Wissenschaftler; sie entwickeln Theorien, stellen Hypothesen auf, testen sie und überarbeiten oder verlassen ihre Theorien nach Bedarf.

Unterrichten kann helfen zu lernen, wenn es die Menschen wirklich unterstützt und befähigt, das zu tun, was sie wollen, wenn es ihnen hilft, herauszufinden, was sie herauszufinden versuchen - aber nur, wenn eine solche Intervention gewünscht, gefordert, eingeladen oder in irgendeiner Weise vom Lernenden akzeptiert wird..⁴

Erleichterung von Programmieren und Offline-Aktivitäten

Machen und Gestalten kann eine Herausforderung sein. Um Kreativität und Innovation zu fördern, müssen wir den Schülern helfen, diese Herausforderungen als einen normalen Teil des Prozesses zu sehen sodass sie lernen, ihr Leben erfolgreich zu gestalten. Um Programmieren zu unterrichten, hilft es, die wichtigsten Phasen des Prozesses zu verstehen, die Reaktionen der Schüler in jeder Phase, hilfreiche Antworten der Lehrer und die vermittelten Fähigkeiten.

Dieser Ansatz kann die traditionelle Denkweise im Klassenzimmer verändern, bei der Lehrer ihr Wissen auf eine weniger traditionelle Art und Weise weitergeben und zu einem Moderator wird, der Schüler zum Lernen anleitet.

Eine zusammenfassende Bewertung des erlernten Wissens kann sich in eine Reihe kleinerer Bewertungen verwandeln, bei denen die von den Schülern erstellten Produkte aufeinander aufbauen und in ein umfangreicheres Abschlussprojekt münden.

Während der Unplugged und Coding-Aktivitäten werden Kinder und Jugendliche dazu angeregt, einige Dinge auszuprobieren und zu sehen, ob sie funktionieren. Versuch und Irrtum können zum Scheitern, aber auch zu Iteration und Innovation führen.

Alle Schüler können herausgefordert werden und wachsen, und dass wir bei der Verfolgung dieses Ziels einen sicheren Raum schaffen, in dem Versagen eine grundlegende Möglichkeit ist. Ein fehlgeschlagenes Experiment ist kein Ende, wie bei einer standardisierten Bewertung, sondern eine Herausforderung, es erneut zu versuchen.

⁴ Was wir mit LERNEN meinen, lego dacta

<http://learn.media.mit.edu/lcl/resources/readings/what-we-mean-by-learning.pdf>, konsultiert am 26.03.2018

Wenn die Schüler frei sind, zu erfinden und zu erschaffen, beginnen sie, Technologie als Mittel zur Lösung realer Probleme zu sehen und ihr Lernen auf die nächste Stufe zu heben.

Unsere Ressourcen sind speziell als praktische Lernwerkzeuge konzipiert, um den Studenten von heute dabei zu helfen, Fähigkeiten für die kreative und digitale Wirtschaft aufzubauen - kritisches Denken, Zusammenarbeit, Kommunikation, Neugier, Problemlösung und Erfindung.

Muss ich als Lehrer wissen, wie man vor Beginn der Aktivitäten kodiert?

Zunächst werden die Ausbilder die Jugendlichen und ihre Lehrer in den Grundlagen der Programmierung schulen. Die Trainer kommen für 10 Stunden im Klassenzimmer. Danach gehen die Schüler der Sekundarschule in die Grundschule.

Die Lehrer der Grundschulen werden ausgebildet, um den Schülern gemeinsam mit den Jugendlichen die Arbeit zu erleichtern. Sie müssen also vor Projektbeginn nicht wissen, wie man programmiert. Sie sollten jedoch bereit sein, mit einem neuen Lernansatz im Unterricht zu experimentieren. Wenn ein Schüler früher fertig ist, laden Sie ihn ein, entweder sein Projekt zu verbessern (indem Sie eine Punktzahl hinzufügen, die Schwierigkeit erhöhen, das Thema des Spiels ändern...) oder ein zweites, schwierigeres zu erstellen.

Am Ende sollten Sie 10 Minuten einplanen, damit die Schüler die Projekte des anderen testen können und eine kurze Diskussion darüber führen können, was sie bisher gelernt haben.

HINTERGRUND

Während des Brüsseler Capital Digital-Projekts waren es die Schulen, die kostenloses Programmieren ermöglichten, da die Materialien und die Hardware zum Unterrichten zur Verfügung gestellt wurden. Es ist wichtig, dass die Trainer, die das Projekt durchführen, eine gute und strukturierte Vorbereitung mit den Lehrkräften und Schulen haben, um die Lernergebnisse und Zeitpläne für die Schüler vor Beginn des Projekts optimal aufeinander abzustimmen.

Lehrkräfte

Die Lehrkräfte haben das Projekt Capital Digital unterstützt, wenn sie nachvollziehen konnten, um welche Aktivitäten es geht, und dass die Schüler begeistert sind und Spaß an den Aktivitäten haben.

Es wurde auch festgestellt, dass Schulleiter stolz darauf waren, Programmier-Workshops im Klassenzimmer anzubieten: Programmieren ist gefragt und Eltern befürworten es mehrheitlich, dass Programmieren in der Schule eingeführt wird. Die Einführung trug dazu bei, das Ansehen der Schule zu erhöhen. Dies ist ausgezeichnet für Schulen, die sich in sozialen Brennpunkt-Gebieten befinden. Es ist wichtig, dass die Trainer vor Beginn des CODINC-Projekts in der Schule mit dem Direktor sprechen, um seine Unterstützung in schriftlicher Form durch eine Vereinbarung zu sichern.

Sag nein zum Chaos im Klassenzimmer, bring etwas Struktur mit!

Lehrkräfte möchten kein Chaos im Klassenzimmer, denn in ihren Augen ist mangelnde Vorbereitung ein Hindernis für den Unterricht. Deshalb müssen die Trainings sehr strukturiert durchgeführt werden. Es ist wichtig, darüber nachzudenken, wie Sie die Workshops organisieren und wie die verschiedenen Übungen geplant werden. In der Vorbereitungsplanung sollte das mit den Jugendlichen besprochen werden. Kann die Struktur in der Klasse und im Klassenzimmer (zum Beispiel, wie die Tische angeordnet sind) für diesen Tag geändert werden oder nicht?

Wie kann ich Gruppenarbeit mit Kindern machen, ohne die Struktur des Klassenzimmers zu verändern? Gibt es einen anderen Raum, der für die Aktivitäten genutzt werden kann?

Dies ist eine wichtige Sache, die vor Projektbeginn mit den Lehrern zu besprechen ist.

Besprechen Sie die Zeitpläne, bevor Sie mit dem Projekt beginnen.

Für die Programmier-Workshops ist es interessant, einige Blöcke von zwei bis drei Zeitstunden zu haben

5 Blöcke à 2 Stunden oder 2 Blöcke à 3 und 2 Blöcke à 2 sind nach unserer Erfahrung ideal. In der Grundschule ist es einfacher, dies zu planen als in der Sekundarstufe.

Machen Sie einen Probelauf, bevor Sie mit dem Projekt beginnen.

Stellen Sie die Programmieraktivität im Klassenzimmer selbst vor und erzählen Sie den Schülern, was sie in diesem Projekt erwartet und wie das Projekt geplant ist.

Lassen Sie sie etwas Praktisches ausprobieren, was ihnen Spaß macht. Die meisten Schülerinnen und Schüler freuen sich, wenn es Projekte im Klassenzimmer gibt, es bietet eine Abwechslung zum üblichen Tagesablauf.

Fragen Sie sie nach ihren Vorbehalten zur Arbeit mit den Grundschul-Kindern und diskutieren Sie diese mit ihnen.

Vorbilder für kleine Geschwister

In den Familien der Kinder sind die Bindungen zwischen den Geschwistern oft sehr stark und "große" Geschwister helfen den Kleinen gerne. Sprechen Sie davon, ein Vorbild für die Kleinen zu sein und über ihre Rolle, die sie dabei spielen können, dass auch jüngere Kinder Programmieren lernen können.

Anhang 1 Darstellung der Lernergebnisse

ARBEITSBLÄTTER

https://docs.google.com/document/d/1wEyC7jfkImp7ks15qr_cx7pFo7kqc5Y0X0lhqYg5U6c/edit#

LINKS

Lightbot (Online und App) - <http://lightbot.com/hour-of-code.html>

Hour of Code: Star Wars - <https://code.org/starwars>

Hour of Code: Minecraft - <https://code.org/minecraft>

WERKZEUGE

Artikel	Geschätzter Preis / Artikel	Anzahl	Geschätzter Gesamtpreis €
Makey Makey Set	Rund 55 Euro für ein Set	10	550
Computer / Laptops		10	
Tablets (optional)		Nach Bedarf (optional)	
Bastelmaterial		Nach Bedarf	
Schreibmaterial		Nach Bedarf	

ANHANG I: REFERENZEN / QUELLEN

Ananiadou, K. (2009). *Fähigkeiten und Kompetenzen des 21. Jahrhunderts für Lernende des neuen Jahrtausends in den OECD-Ländern.*

Brennan, K., & Resnick, M. (2012). *Mit artefaktbasierten Interviews die Entwicklung des computergestützten Denkens im interaktiven Mediendesign untersuchen.* Vortrag auf der Jahrestagung der Educational Research Association, Vancouver, BC, Kanada.

Stiftung Raspberry Pi. (2017). *CodeClub Lehrplan.* Abgerufen am 6. März 2018 von <https://codeclubprojects.org/en-GB/curriculum/>

<http://curriculumvandetoekomst.slo.nl/21e-eeuwse-vaardigheden>

Die zweite digitale Kluft in Europa. Eine länderübergreifende Studie über die digitalen Lese- und Navigationsfähigkeiten der Schüler* Davide Azzolinia†, Antonio Schizzerottoa,b

<https://irvapp.fbk.eu/wp-content/uploads/2017/09/FBK-IRVAPP-Working-Paper-No.-2017-02.pdf>