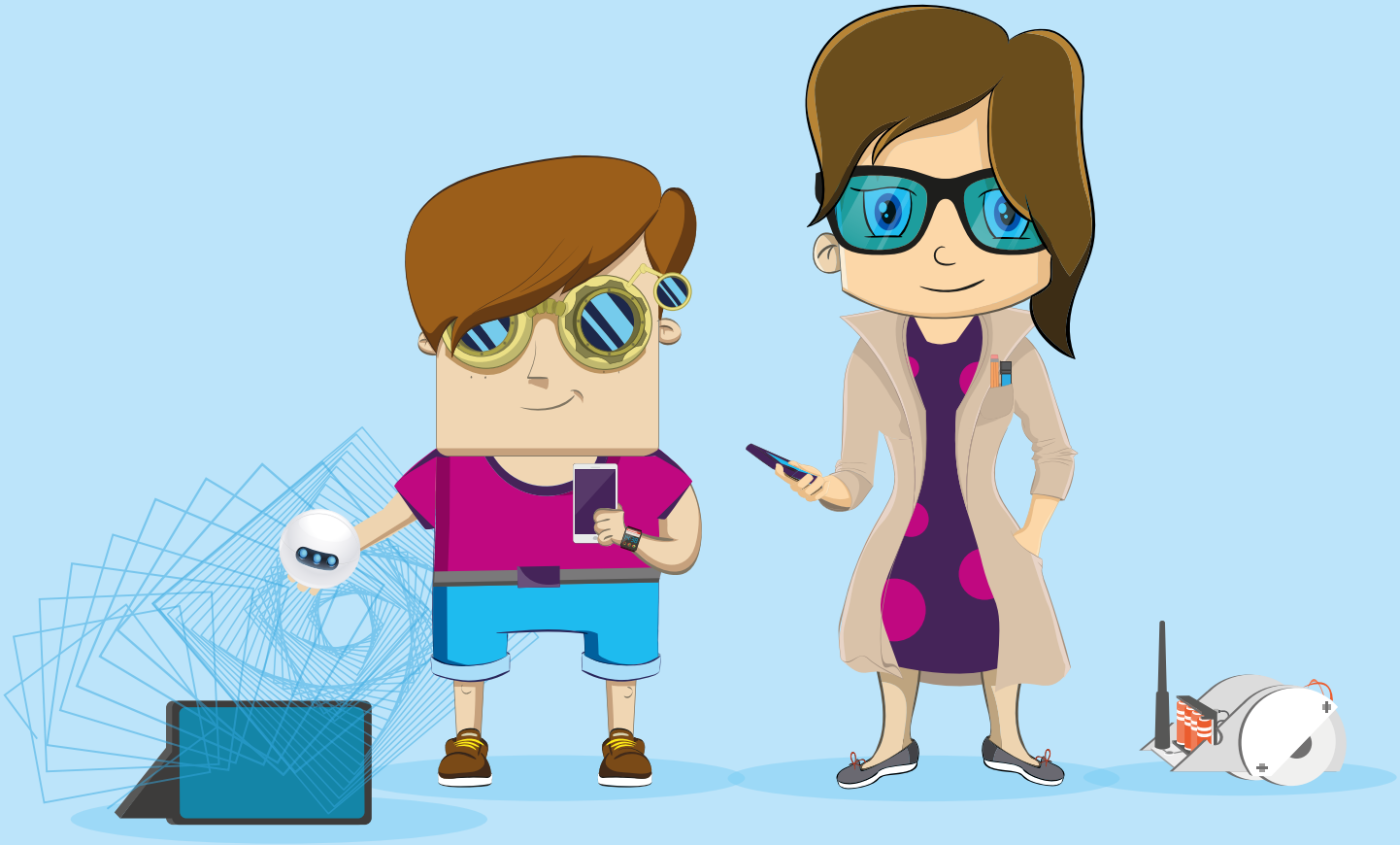
CODE your life

Programmieren für die Zukunft

Ein Handbuch für das Programmieren mit Kindern



www.code-your-life.org

Microsoft YouthSpark

Inhalt

Impressum

Herausgeber: Förderverein für Jugend und Sozialarbeit e. V.
Verantwortlich: Thomas Schmidt
Konzeption und Umsetzung: Helliwood media & education, Marchlewskistr. 27, 10243 Berlin, www.helliwood.com
Redaktion: Jutta Schneider, Jörg Peleikis, Thomas Schmidt, Jacqueline Graf, Christine Schulz
Gestaltung und Satz: Christiane Herold
Bildnachweis: Christian Griebel, Christiane Herold, shutterstock.com und A. Bedoy für die Initiative D21

Berlin 2016

Code your Life - Programmieren für die Zukunft

> Das Verständnis der Informatik und der Logik von Algorithmen als der Sprache der digitalen Welt ist für einen selbstbestimmten Umgang mit der Digitalisierung in der Alltags- und Berufswelt von herausragender Bedeutung.
>
> Deutscher Bundestag

Die Initiative Code your Life stellt sich einer der drängendsten Herausforderungen der Zukunft: jungen Menschen eine aktive Teilhabe an der digitalisierten Gesellschaft zu ermöglichen. Mit diesem Ziel führt Code your Life Kinder ab zehn Jahren an das Programmieren heran und lässt sie hinter die Kulissen der digitalen Welt schauen.

Die Kinder erfahren, wie das Schreiben von Computercodes funktioniert und was mit neuen Technologien alles möglich ist. Die wichtigste Botschaft dabei: Programmieren ist kreativ, bunt, vielfältig und macht Spaß. Deshalb nutzt Code your Life Methoden und Tools, die niedrigschwellig, aber zugleich herausfordernd sind und immer sofort sichtbare Ergebnisse hervorbringen. Wow-Effekte sind dabei garantiert und machen Lust auf mehr. So steuern die Kinder beispielsweise kleine Miniroboter und lassen digitale Schildkröten Kunstwerke zeichnen, bauen Twitterwalls und programmieren leuchtende Botschaften auf LED-Boards. Die Kinder werden bei Code your Life zu aktiven Produzentinnen und Produzenten ihrer Medien und lösen sich von der Rolle der passiv Rezipierenden.

Warum machen wir, was wir machen?

Zweifellos schreitet die Digitalisierung unserer Gesellschaft mit großen Schritten voran und Programmcodes durchziehen und prägen inzwischen viele Bereiche unseres Lebens. Tagtäglich nutzen wir ganz selbstverständlich Dinge, die erst durch das Schreiben von Computercodes möglich gemacht wurden. Sei es das Buchen einer Zugfahrkarte, die Nachricht an den besten Freund, das Fotografieren mit dem Smartphone oder oder oder... Eine Welt ohne Programmierung ist heute undenkbar geworden.

Kinder wachsen in dieser Welt auf und kommen bereits in frühen Jahren mit neuen Technologien und einer Vielzahl von Medien in Berührung. Wer diese Welt auch in Zukunft verantwortungsvoll und aktiv mitgestalten will, braucht ein Verständnis für Zusammenhänge und Hintergründe. Nicht ohne Grund fordern Vertreterinnen und Vertreter aus Wirtschaft und Politik, Programmieren als zweite Fremdsprache in Schulen zu etablieren.

1. Antrag der Fraktionen der CDU/CSU und SPD: „Durch Stärkung der Digitalen Bildung Medienkompetenz fördern und digitale Spaltung überwinden". Drucksache des deutschen Bundestages 18/4422

Kompetente Teilhabe

Um an einer digitalisierten Welt kompetent teilhaben zu können, braucht es einen Blick dafür, wie digitale Technologien funktionieren. Das Erkunden, Ausprobieren und Verstehen von Programmiercodes ist ein erster wichtiger Schritt in diese Richtung.

Junge Menschen für Informationstechnologien und Programmieren zu begeistern, wird daher eine zunehmend bedeutsame Rolle spielen. Schon jetzt mangelt es der Wirtschaft an Fachkräften im IT-Sektor und der Bedarf wird noch weiter steigen.

Die Sicherung von Nachwuchs in den am schnellsten wachsenden neuen Arbeitsfeldern ist ein Schlüsselthema für Politik und Wirtschaft in Deutschland. Um dieser notwendigen Nachwuchssicherung gerecht werden zu können, ist es besonders wichtig, Kinder und Jugendliche schon früh an die Informationstechnologien und das Programmieren heranzuführen. Dabei geht es jedoch nicht nur darum, zu lernen, wie Algorithmen geschrieben werden, sondern auch um die Entwicklung von Schlüsselkompetenzen des 21. Jahrhunderts wie Problemlösekompetenz, Reflexionsfähigkeit, Verantwortungsbewusstsein und ganzheitliches Denken.

Programmieren als Zukunftskompetenz

Zweifellos sind Coding-Kenntnisse bereits heute von zentraler Bedeutung – und geradezu unerlässlich für die Zukunft junger Menschen und für die Zukunft des Industriestandortes Deutschland.

Code your Life möchte allen Kinder ermöglichen, diese Kompetenzen zu entwickeln, unabhängig von Herkunft, Bildungshintergrund, Geschlecht oder sozialem Status.

Die Angebote der Initiative richten sich daher ganz bewusst schon an Zehnjährige, denn Kinder in diesem Alter sind besonders neugierig, voller Entdeckerlust und sehr aufgeschlossen gegenüber neuen Technologien, Geräten und Anwendungen. Und das gilt für Mädchen genauso wie für Jungen.

Stereotype wie „Das ist doch nur was für Jungs“ haben sich noch nicht verfestigt. Die Kinder sind unvoreingenommen und stellen sich mit Freude kleinen und großen Herausforderungen. Diese Unvoreingenommenheit ist die beste Voraussetzung, um Mädchen und Jungen gleichermaßen an das Programmieren heranzuführen.

Mädchen und Frauen in der IT

Noch immer ist der Anteil von Frauen in der IT-Branche sehr gering. Es gibt zum Beispiel deutlich mehr männliche (circa 85 Prozent) als weibliche Studierende (circa 15 Prozent) der Informatik.[2]

Code your Life ist auch der Versuch, dieses Verhältnis zu ändern und etwaige Berührungsängste mit dem Programmieren bei den teilnehmenden Mädchen gar nicht erst aufkommen zu lassen.

Wer sind die Initiatoren von Code your Life?

Code your Life ist eine Initiative des 21st Century Competence Centers des Fördervereins für Jugend und Sozialarbeit e. V.

Der Förderverein übernimmt als Kompetenzträger für digitale Kultur, Bildung und Gemeinwesen in Deutschland eine hohe Verantwortung, wenn es darum geht, Bildungschancen für junge Menschen zu erhöhen und sie zu einer erfolgreichen Teilhabe an der digitalisierten Gesellschaft zu befähigen. Bereits seit 25 Jahren engagiert sich der Förderverein mit innovativen Ideen und Projekten für die

2 Bundeszentrale für politische Bildung „Zahlen und Fakten. Die soziale Situation Deutschlands“ Quelle www.bpb.de/nachschlagen/zahlen-und-fakten/soziale-situation-in-deutschland/61669/studierende, Zugriff am 06.08.2015

Entwicklung und Förderung einer weitreichenden Medienkompetenz bei Kindern und Jugendlichen.

Mit der Initiative Code your Life konnte nun ein langgehegter Wunsch verwirklicht werden: das Programmieren in die Schulen und zu den Kindern zu bringen und regelmäßig mit jungen Menschen Programmcodes zu schreiben. Die erforderlichen Kompetenzen dafür sind im Förderverein versammelt, in dem ein interdisziplinäres Team aus Pädagoginnen und Pädagogen, Redakteurinnen und Redakteuren sowie Entwicklerinnen und Entwicklern arbeitet.

Code your Life wird im Rahmen des weltweiten Programms YouthSpark von Microsoft unterstützt. Microsoft engagiert sich bei der Initiative mit dem Ziel, allen jungen Menschen den Zugang zu digitalen Technologien zu ermöglichen und so deren Zukunftschancen zu erhöhen. Diese Unterstützung war die Initialzündung für Code your Life und hat diese Initiative erst möglich gemacht.

Was passiert bei Code your Life?

Im Dezember 2014 startete die Initiative mit einer sechsmonatigen Pilotphase an Modellschulen. Mit ausgewählten Lerngruppen konnte so in regelmäßigen Workshops erprobt werden, woran Kinder Freude haben, welche Methoden und Tools sich für das Programmieren mit Kindern eignen und inwiefern die Kinder dabei eigene Problemlösungsstrategien entwickeln können.

Herausgekommen ist ein Curriculum mit Lerneinheiten, das von der ersten Annäherung an einen Programmierwortschatz über einfache Programmierprinzipien bis hin zu komplexen Aufgabenstellungen eine stimmige Einführung ins Programmieren ermöglicht. Ein Auszug des Curriculums findet sich auch in diesem Handbuch. Ergänzend dazu gibt es Tipps und Tricks für geeignete Tools und Anwendungsszenarien im Unterricht und in außerschulischen Lernkontexten.

Die Pilotphase zeigte ganz deutlich: Kindern macht Programmieren Spaß. Sie verstehen, wie Programmcodes funktionieren, interessieren sich für das, was dabei passiert, und fangen ohne Berührungsängste ganz selbstverständlich an zu programmieren. Und auch die pädagogischen Fachkräfte zeigten großes Interesse und großes Engagement und äußerten vielfach den Wunsch nach Begleitung und unterstützendem Material. Ebenso interessiert und engagiert waren die Politik sowie Bildungsexpertinnen und -experten. So gab es für die Initiative zum Beispiel bereits eine Einladung ins Bundeskanzleramt und einen Dialog mit der Bundeskanzlerin.

Die positiven Erfahrungen der Pilotphase zeugen von einem vielversprechenden Anfang und sprechen für den Ansatz der Initiative: Programmieren für die Zukunft.

Wie wir machen was wir machen!

Programmieren macht Spaß. Genau das vermitteln wir bei Code your Life. Wir wollen die Vorstellungskraft der Kinder wecken und sie für kreative Problemlösungen begeistern. Mit unseren Tools und Anwendungen schaffen wir Situationen, in denen sie bereits nach kürzester Zeit erstaunliche Dinge programmieren können. Wir vertrauen dabei auf die Stärken der Kinder und ihre intuitiven Fähigkeiten und ermöglichen ihnen Erfolgserlebnisse, die sie vorher nicht erwartet hätten.

Für den Einstieg ins Programmieren greifen wir ganz bewusst auf die Programmiersprache Logo zurück, die bereits Ende der Sechziger Jahre von Seymour Papert entwickelt wurde, um Programmieren in Bildungskontexten effektiv einzusetzen. Insbesondere die Turtle-Graphiken eignen sich wunderbar für ein

schrittweises Erkunden von Programmierprinzipien.

> The role of the teacher is to create the conditions for invention rather than provide ready-made knowledge.
>
> Seymor Papert

Hier „übersetzt" eine virtuelle Schildkröte die eingegebenen Codezeilen in Bewegung und zeichnet dabei ihren Weg nach.

Wie Seymour Papert glauben auch wir daran, dass Kinder am besten lernen, wenn sie das Wissen nicht vorgesetzt bekommen, sondern es sich selbst aufbauen können. Das angewandte Selbstwirksamkeitsprinzip führt zu Erfolgserlebnissen, welche die Lernfortschritte verstärken. Dahingehend geben wir in unseren Lernszenarien den richtigen Lösungsweg nicht vor, sondern ermöglichen den Kindern und Jugendlichen, Herausforderungen im Team mit unserer Unterstützung selbst zu bewältigen.

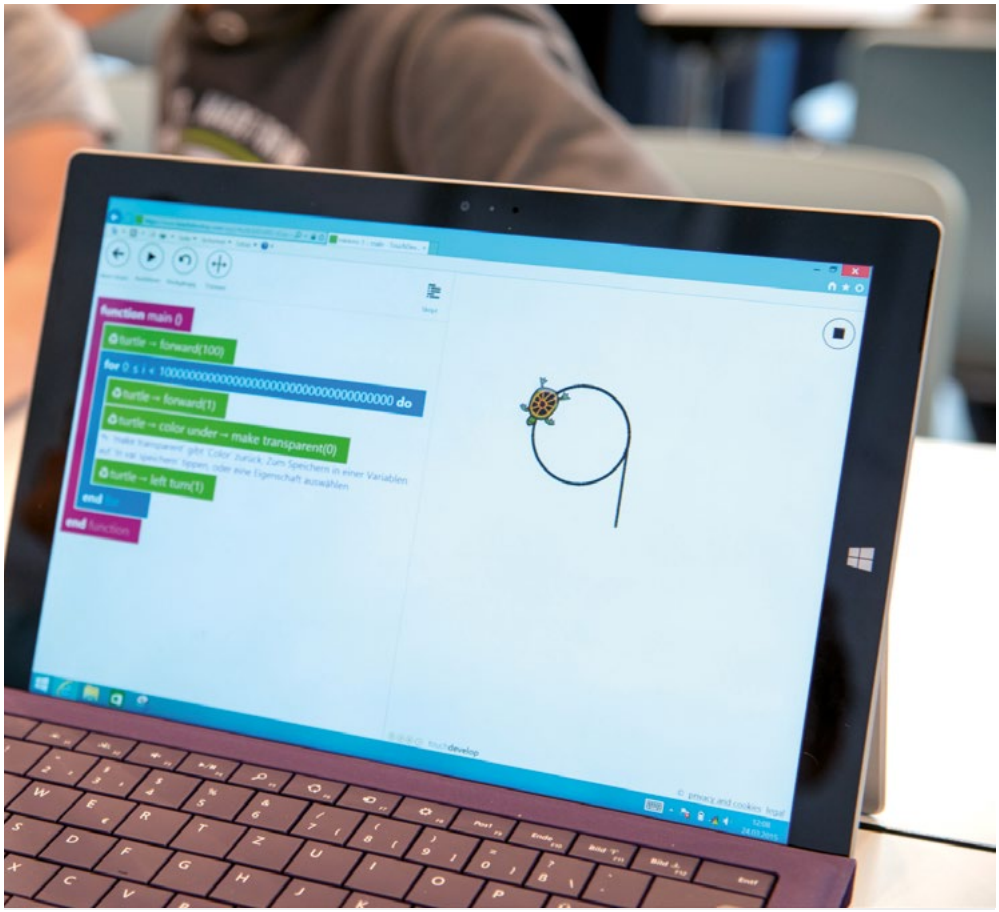
In den Lernszenarien beginnen die Kinder damit, erst einmal einfache Befehle untereinander zu schreiben und ein Programm immer wieder auszuführen. Sie erkennen, dass sie durch ein Hinzufügen von Codezeilen oder der Veränderung von Parametern das Ergebnis beeinflussen können. Irgendwann entwickelt sich bei den Kindern das Bedürfnis nach weiteren strukturgebenden Elementen, so dass Programmierprinzipien wie Schleifen und Variablen eingeführt werden können. Dabei lernen die Kinder die Dinge genau dann, wenn sie sie für die Problemlösung benötigen.

Wir verzichten ganz bewusst auf Blockprogrammieren (wie z.B. Scratch) und lassen die Kinder gleich zu Beginn tatsächlich Codezeilen schreiben. So lernen die Kinder, dass für das Programmieren eine eigene Sprache notwendig ist, die einen eigenen Wortschatz hat und nach bestimmten Regeln, der Syntax funktioniert. Die Kinder entdecken die Programmiersprache explorativ durch Schreiben von Code und immer wiederkehrende kleine Herausforderungen. Haben die Kinder dies verinnerlicht, gelingt später der Transfer des Wissens in visuelle Programmierumgebungen mit Block-Formaten ohne Probleme.

Zur Unterstützung und Verinnerlichung des Gelernten legen wir das Tablet auch immer wieder weg und simulieren das Computerprogramm im Raum. Die Kinder „spielen" die Turtle und laufen die geforderten Figuren selbst im Raum ab. Dadurch fällt es ihnen leichter, die richtigen Winkel und Richtungen zu verstehen, sie müssen genau überlegen wie weit sie sich drehen müssen und wie viele Schritte zu gehen sind.

Haben sie diese Prinzipien verstanden und ist die Begeisterung für Computer Science geweckt, ist der Weg bereitet, um auch komplexere Sachen zu programmieren und ihre Umwelt mit neuen Ideen und Lösungen zu bereichern.

Curricula



Programmieren mit Kindern und Jugendlichen kann man auf unterschiedlichste Weise angehen. Im Rahmen der Pilotphase von Code your Life wurden verschiedene Lernszenarien, Tools und Herangehensweisen ausprobiert und evaluiert. So entstanden mit der Zeit ausführliche Curricula, die sowohl von Lehrkräften als auch von Schülerinnen und Schülern mit Begeisterung aufgenommen werden. Im vorliegenden Handbuch ist ein Auszug dieser Curricula zu Turtle-Grafiken mit Logo zu finden.

Sie beinhalten vollständige Unterrichtseinheiten mit einem detailliert beschriebenen Unterrichtsablauf, weiterführenden Tipps oder Verweise auf didaktische Prinzipien. Die Unterrichtseinheiten bauen aufeinander auf und können als ein in sich geschlossenes Unterrichtskonzept für einen Coding-Lehrgang verwendet werden.

Die Unterrichtseinheiten eignen sich insbesondere für Kinder, die noch keine Vorerfahrung im Programmieren haben.

Die Turtle

„Wir haben es geschafft!" Diesen freudigen und stolzen Ausruf hört man bei der Durchführung des ersten Curriculums häufig.

Die Unterrichtseinheit „Die Turtle" bietet den Kindern den Einstieg in die Thematik des Programmierens und beinhaltet sogleich Erfolgserlebnisse. Es wird darüber gesprochen, was ein Computerprogramm ist, welche Sprache verwendet wird und wie Befehle gegeben werden.

Die Kinder lernen die Turtle und die Programmierumgebung TouchDevelop kennen. Intuitiv fangen sie an, zu experimentieren und Ideen zu entwickeln.

Und siehe da: Schon während dieser ersten Stunde schreiben die Kinder ihre ersten Codezeilen und entwerfen kleine Programme. Auch auf die ersten Schwierigkeiten werden sie stoßen und diese gemeinsam meistern.

Überblick

Kompetenzen

Zeitaufwand	90 Minuten
Technik	touchfähige Geräte, WLAN (auch am PC mit LAN)
Methoden	Gruppenarbeit, Frage und Antwort, Simulation
Vorkenntnisse	Keine notwendig

Die Schülerinnen und Schüler ...

- ¬ lernen Logo als Bestandteil der Programmierumgebung www.touchdevelop.com als eine erste Programmiersprache kennen.
- ¬ begegnen der Turtle und verstehen, wie sie die Turtle bewegen können.
- ¬ erarbeiten grundlegende Befehle in Logo.
- ¬ wissen um die Bedeutung von Parametern.
- ¬ wissen am Ende, wie sie es schaffen, dass die Turtle ein großes buntes Quadrat auf den Bildschirm zeichnet.

Ablauf

Phase	Aufgabe	Methode	Zeit
Sensibilisierung	Was ist Programmieren?	F & A	5‘
Vorbereitung	Erarbeitung grundlegender Befehle und Parameter	Simulation	15‘
Arbeitsphase	Der Strich	Gruppenarbeit & Simulation	60‘
	Das einfache Quadrat		
	Das bunte Quadrat		
	Das große Quadrat		
Ausblick	Mäander	Hausaufgabe	10‘

1

Unterrichtsverlauf

Vorbemerkung

Die Lerneinheiten von Code your Life beinhalten immer wieder kleine Situationen, in denen die Lehrkraft mit den Kindern „offline" sowohl Programmierer als auch das Computerprogramm simulieren. Gehen Sie daher sehr spielerisch an das Programmieren heran und scheuen Sie sich nicht, vor den Kindern die Schildkröte (natürlich abstrahiert) zu spielen. Den Kindern wird es helfen, bestimmte Programmierprinzipien zu verstehen, wenn sie es erst einmal anschaulich im Klassenraum erleben. Wenn die Kinder es geschafft haben, Sie als Schildkröte offline zu steuern, wird es ihnen auch gelingen, die Turtle auf dem Bildschirm zu bewegen.

Ausprobieren lassen! Kinder sind in diesem Alter geprägt von einer großen Neugier gegenüber neuen Medien, Computern und Smartphones. Die Motivation, die Rätsel der Turtle zu lösen, ist groß. Lassen Sie die Kinder unbedingt so viel wie möglich selbst herausfinden! Auch wenn sie manchmal zuerst auf der falschen Fährte sind, werden sie dies sehr schnell merken. Gemeinsam können die Kinder (mit Ihnen) auf Fehlersuche gehen. Geben Sie also nicht zu viel vor, sondern geben Sie die richtigen Hilfestellungen.

Je nachdem wie unterschiedlich das Lernniveau der Klasse ist, können Sie die Aufgaben der Arbeitsphase auch individuell an schnellere Gruppen bereits herausgeben. Lassen Sie gegebenenfalls die Teams sich untereinander Unterstützung geben.

Phase 1 | Sensibilisierung

Wo spielt Programmierung eine Rolle? Je nach Vorkenntnissen der Schülerinnen und Schüler geht es zu Beginn der Unterrichteinheit darum, ein gemeinsames Verständnis einzuholen, was Programmieren eigentlich ist. Fragen Sie Ihre Lerngruppe, wo Programmieren im Alltag überall eine Rolle spielt, ob sie selbst schon einmal programmiert haben, was man dazu braucht etc.

Phase 2 | Vorbereitung

Das Ziel der Vorbereitungsphase ist die Erarbeitung eines grundlegenden Wortschatzes für die Turtle, sowie ein Verständnis für die Funktionsweise von Parametern zu schaffen. Die Kinder sollen verstehen, wie eine Zeile Code aufgebaut ist und wie sie die Turtle bewegen können.

2.1 Schritt eins | Die grundlegenden Befehle

Spielen Sie die Schildkröte! Führen Sie die Turtle ein, indem Sie selbst eine Schildkröte simulieren. Stellen Sie sich dazu vor die Schüler und strecken Sie die Arme nach vorn. Die Arme sind die „Nase" der Turtle und geben die Richtung an. Lassen Sie die Kinder Ihnen Befehle geben, und führen Sie die Befehle aus. Achten Sie darauf, wirklich genau das auszuführen, was die Kinder sagen. Sollen Sie „Laufen", gehen Sie auf der Stelle und zwar so lange bis die Kinder „Stopp" rufen. Fragen Sie dann, welchen Befehl sie geben würden, damit Sie auch von der Stelle loskommen. Kommt der richtige Befehl, laufen Sie dann natürlich wieder immer weiter geradeaus, bis es nicht mehr weitergeht.

So werden nach und nach folgende Begriffe von den Schülern gefunden. Notieren Sie die Befehle an der Tafel.

Hinweis Keine Scheu vor den englischen Begriffen. Die Kinder verstehen sofort, dass die Turtle als „Computertier" eine fremde Sprache (hier Englisch) spricht.

```
turtle → forward (200)
turtle → back (100)
```

2.2 Schritt zwei | Parameter

Drehen Sie sich im Kreis! Simulieren Sie wieder die Turtle und fragen Sie nach, wie man die Turtle dazu bekommen könnte, sich in eine andere Richtung zu bewegen. Geben Sie eventuell den Tipp, dass es sich dabei um eine Grad-Angabe handelt. Drehen Sie ich als Turtle im Kreis und lassen die Kinder bei 90° und bei 180° Stopp rufen. Ergänzen Sie das Tafelbild um die neuen Befehle und ergänzen Sie die Klammern für die Parameter.

```
turtle → left turn (90)
turtle → right turn (180)
```

Nun geht es um die Schrittlänge. Zeigen Sie, wie Sie Schritt für Schritt als Turtle vorwärts laufen. Erklären Sie, dass ihre Fußlänge, einem Pixel auf dem Bildschirm entspricht. Zeigen Sie, was Pixel sind!

Hinweis Um Schwierigkeiten vorwegzunehmen und den Fluss der Erarbeitung jetzt hier nicht zu stören, verlagern Sie den Aufbau eines Bildschirms aus Pixeln gerne in eine vorbereitende Stunde oder in die Einführung. Dies ist ein anerkanntes und probates

didaktisches Mittel.

Ergänzen Sie wieder die Klammern und Parameter bei den Befehlen. Lassen Sie sich wieder Befehle von den Schülern geben.

turtle → forward (200)
turtle → back (100)

Jetzt haben Sie einen grundlegenden Wortschatz geschaffen, mit dem die Schüler die erste Aufgabe bewältigen können.

In dieser Phase übertragen die Kinder nun das eben Gelernte aus der Simulation auf die tatsächliche Programmierung der Turtle. Schritt für Schritt führen sie verschiedene Aufgaben aus und lösen die ersten Herausforderungen.

Phase 3 | Arbeitsphase

3.1 Gruppen einteilen und Namen finden

An dieser Stelle bietet es sich an, die Lerngruppe in Teams einzuteilen. Wir empfehlen, zwei bis drei Kinder an einem Computer oder Tablet arbeiten zu lassen. So können sie gemeinsam überlegen und sich austauschen, ohne dass zu viele an einem Gerät sitzen.

Lassen Sie die Kinder Namen für ihre Teams finden, wie z.B. „Team Turtlepower". Schreiben Sie die Namen an die Tafel. So können Sie den Überblick behalten, welches Team die nun folgenden Aufgaben schon gelöst hat. Außerden erkennen sie, wie schnell die Lerngruppen sind, um gegebenenfalls das Lerntempo anzupassen..

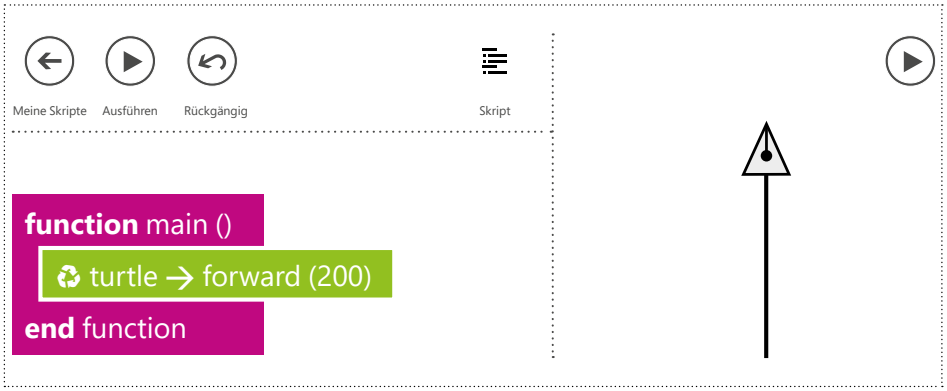
Hinweis Wir empfehlen, in dieser Phase noch ohne Logins auf www.touchdevelop.com zu arbeiten und das erste Programm einfach so zu benennen, wie die Gruppe heißt.

3.2 Der Strich

Jetzt dürfen die Kinder die Geräte anschalten. Erläutern Sie den Kindern, wie Sie auf www.touchdevelop.com zur Turtle kommen. Lassen Sie die Kinder beim Skriptnamen, den Teamnamen eintragen. Eine Handreichung hierzu finden Sie unter www.code-your-life.org.

Geben Sie die erste Aufgabe:

!! Aufgabe:
Die Schildkröte soll einen geraden Strich mit einer Länge von 200 Pixeln zeichnen!



i Hinweis! Erklären Sie nun nicht zu viel, sondern lassen Sie die Kinder die Bedienoberfläche der Programmierumgebung autodidaktisch kennenlernen.

Hat eine Gruppe einen Strich geschafft, notieren Sie dies an der Tafel.

3.3 Das einfache Quadrat

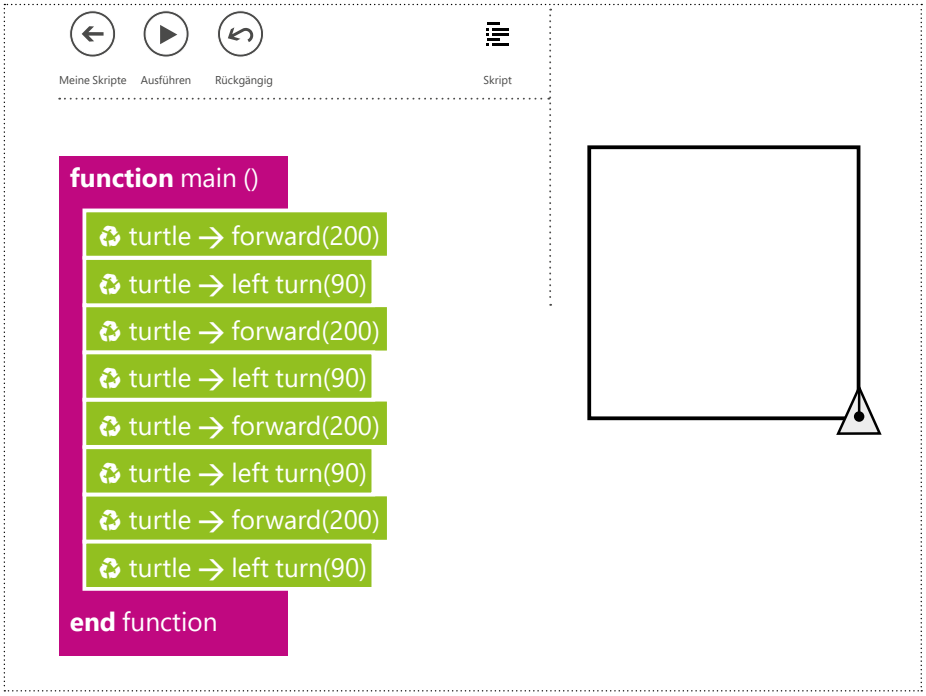
Nun sind sie Kinder bereit, weitere Codezeilen zu schreiben. Die nächste Aufgabe ist nun, dass die Turtle ein Quadrat zeichnet. Reflektieren Sie mit den Kindern noch einmal, welche Eigenschaften ein Quadrat hat und wie die groß die Winkel sind. Halten Sie dies als Tafelbild fest.

Geben Sie nun die zweite Aufgabe:

!! Aufgabe:
Die Schildkröte soll ein Quadrat mit einer Seitenlänge von 200 Pixel zeichnen.

1

Lösungsbeispiel



CyL 1.3.3 Quadrat

Dokumentieren Sie an der Tafel, wenn ein Team die Aufgabe gelöst hat.

Hinweis! Alle Programme finden Sie unter dem angegebenen Namen in TouchDevelop.com. Geben Sie den Namen einfach im „Hier suchen"-Feld im Bereich „Meine Skripte" ein.

3.4 Das Quadrat in 4 verschiedenen Farben

Haben einige Teams die vorangehende Aufgabe bereits gelöst, können Sie diesen schon die nächste Aufgabe stellen:

Aufgabe:
Die Schildkröte soll wieder ein Quadrat mit eine Seitenlänge von 200 Pixel zeichnen. Diesmal sollen alle Seiten jeweils eine andere Farbe haben.

1

Lassen Sie die Kinder erst einmal selbst überlegen, was die Turtle braucht, um mit verschiedenen Farben zu malen. Geben Sie allerdings den Hinweis, dass der bereits geschriebene Code des einfachen Quadrats nicht gelöscht werden muss.

Sobald alle Teams die vorangehende Aufgabe gelöst haben, holen Sie die Aufmerksamkeit der Klasse wieder nach vorne und nehmen Sie sich dem Problem der unterschiedlichen Farben an.

Spielen Sie nun wieder eine Schildkröte und gehen Sie mit den Kindern folgende Fragen durch:

¬ Was braucht die Schildkröte zum Malen? (Antwort: einen Stift)
¬ Was heißt Stift auf Englisch? (Antwort: pen)

Holen Sie zur Veranschaulichung einen Stift und nehmen Sie ihn in die Hand.

¬ Kann die Schildkröte so bereits zeichnen?
(Antwort: Nein, denn der Stift muss erst abgesetzt werden.)

```
turtle → pen down
turtle → pen up
```

Nehmen Sie folgende Befehle ins Tafelbild mit auf, wir brauchen sie später noch.

¬ Was heißt Farbe? (Antwort: Color)
Schreiben Sie den noch nicht ganz fertigen Befehl an die Tafel.

```
turtle → pen color (...)
```

¬ Wie müsste der Befehl lauten, dass die Turtle eine andere Farbe annimmt?
(Antwort: z.B. Stiftfarbe wechseln) Helfen Sie den Kindern und ergänzen Sie den Befehl!

```
turtle → set pen color (...)
```

1

¬ Was kommt nun in die Klammer? Was ist der Parameter? (Antwort: die Farbe)
Vervollständigen Sie den Befehl für das Tafelbild.

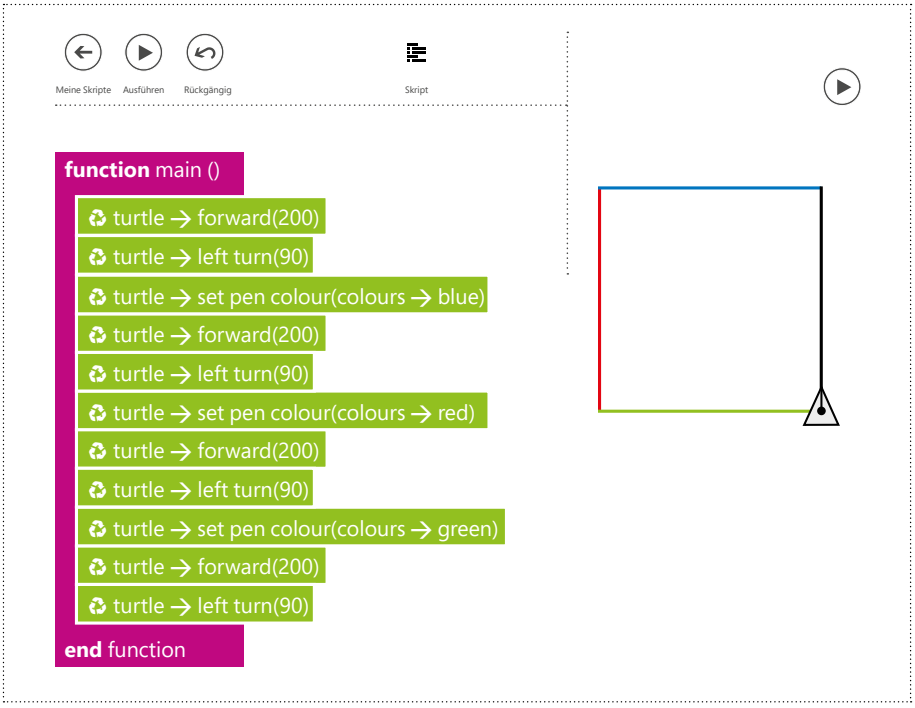
turtle → set pen color (colors → (red)

¬ Wo muss der Befehl stehen? (Antwort: z.B. idealerweise vor jedem TURTLE -> FORWARD)

Hinweis! Laufen Sie gegebenenfalls noch einmal als Turtle das Quadrat ab und halten Sie vier Stifte in der Hand. Die Kinder sollen Ihnen nun zur richtigen Zeit den richtigen Befehl geben.

Lassen Sie die Kinder nun mit der Turtle das Quadrat zeichnen.

Lösungsbeispiel für ein Quadrat mit einer Seitenlänge von 200 Pixel und 4 verschiedenen Farben.



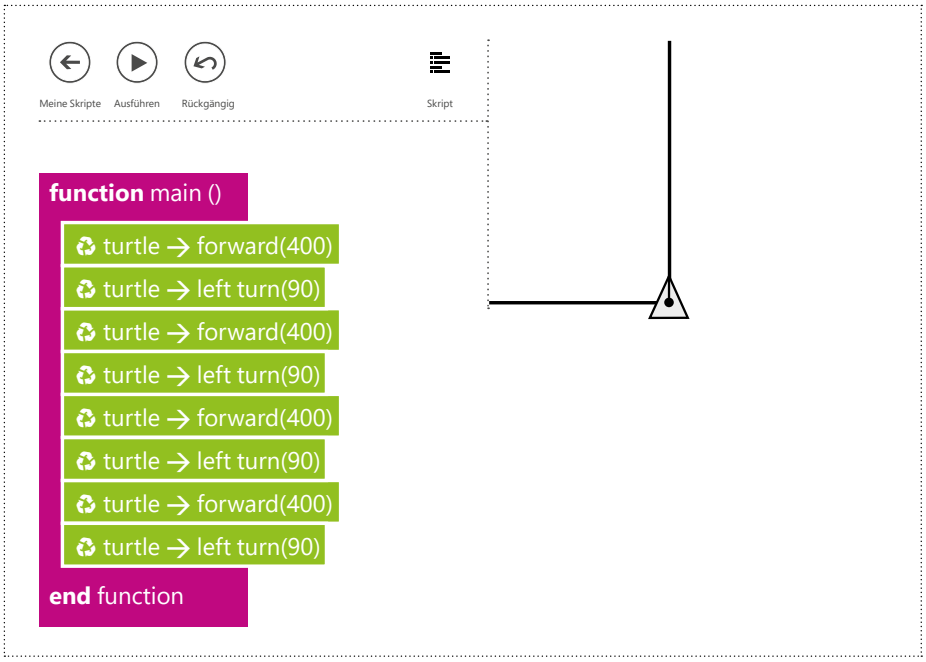
CyL 1.3.4 Buntes Quadrat

3.5 Das große Quadrat

Jetzt bekommen die Kinder die letzte Aufgabe für die heutige Lerneinheit:

!! Aufgabe:
Die Schildkröte soll wieder ein Quadrat malen. Diesmal mit einer Kantenlänge von 400 Pixel. Und das Quadrat soll auf dem Bildschirm in seiner Gänze zu sehen sein.

Die Kinder werden sofort loslegen. Doch was sich leicht anhört, beinhaltet eine kleine Schwierigkeit, welche die Kinder auch umgehend auf ihren Bildschirmen sehen werden: Die Schildkröte läuft aus dem Bild hinaus.



Ermuntern Sie die Kinder, selbst zur Lösung des Problems zu kommen. Gehen Sie von Team zu Team und geben Sie, falls notwendig, kleine Hinweise oder Hilfestellungen.
Es gibt unterschiedliche Wege, welche die Kinder einschlagen, um das Quadrat ganz auf den Bildschirm zu bekommen. Reflektieren Sie dahingehend mit allen gemeinsam:

¬ Wie seid ihr zur Lösung gekommen, wie habt ihr das gemacht?

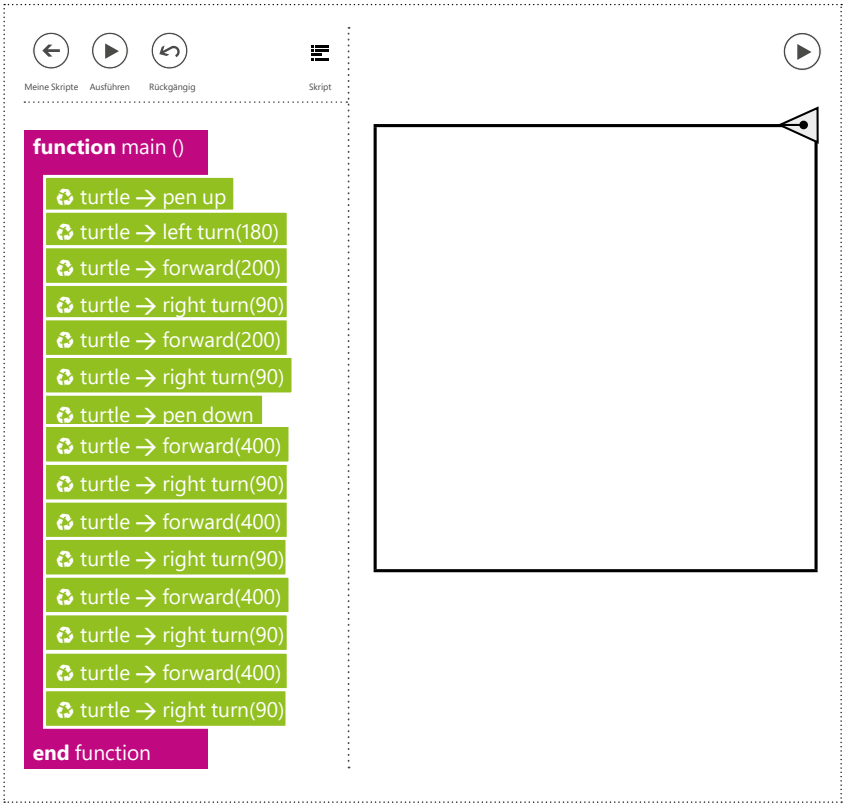
1

Mögliche Lösungswege:

- Über PEN UP und PEN DOWN zu einer Ecke des Bildschirms laufen.
- Weiße Farbe bei SET PEN COLOR wählen.
- Bei nicht geteiltem Bildschirm: erst 200 Pixel nach unten, am Ende des Quadrats ebenso.

Kein Lösungsweg ist falsch! Allerdings sollten die Kinder zum Schluss das Prinzip verstanden haben, dass man die Turtle zu jedem beliebigen Punkt des Bildschirms laufen lassen kann, ohne dass sie malt. Deshalb benötigen sie die Befehle PEN UP und PEN DOWN.

Lösungsbeispiel für ein Quadrat mit der Seitenlänge 400 Pixel, das ganz auf dem Bildschirm zu sehen ist.



CyL 1.3.5 Großes Quadrat

Phase 4 | Ausblick

Haben Sie am Ende der Unterrichtseinheit noch Zeit, lassen Sie die Kinder noch ein wenig frei „spielen“ und Befehle für die Turtle ausprobieren.

In der nächsten Unterrichtseinheit wird es um Mäander gehen. Bitten Sie die Kinder darum, zuhause einmal zu recherchieren, was ein Mäander ist und nach Möglichkeit ein Beispiel für einen Mäander auszudrucken.

Gesamtwortschatz der Turtle (Stand Tutorial 1)	
♻ turtle → pen up	Die Turtle nimmt den Stift hoch.
♻ turtle → pen down	Die Turtle setzt den Stift ab.
♻ turtle → left turn (90)	Die Turtle dreht sich um 90° nach links.
♻ turtle → right turn (180)	Die Turtle dreht sich um 180° nach rechts.
♻ turtle → forward (200)	Die Turtle läuft um 200px nach vorne.
♻ turtle → back (100)	Die Turtle läuft um 100px rückwärts.
♻ turtle → set pen color (colors -> red)	Die Turtle wechselt die Farbe auf „rot“.

Die Bruno-Schleife

Das Ziel der zweiten Unterrichtseinheit ist es, den Kindern das erste strukturgebende Element, das Prinzip der Schleife, nahezubringen. Und zwar, indem die Kinder selbst darauf kommen.

Dazu sollen die Kinder, angefangen bei einem Quadrat weitere regelmäßige n-Ecke programmieren. Je höher das n, desto mehr Codezeilen müssen die Kinder nach ihrem aktuellen Kenntnisstand untereinander schreiben.

„Puh, das ist schon die 30ste Zeile Code. Das muss doch einfacher gehen", werden die Ersten beim 36-Eck stöhnen.

Das Problem ist also gefunden und der Wunsch groß, es zu lösen. Gemeinsam gehen die Kinder auf die Suche und entdecken spielerisch die Möglichkeit, Wiederholungen in ihr Skript einzubauen. Wie von alleine verfügen sie nun über neues Wissen, mit dessen Hilfe sie schon die wunderbarsten Muster programmieren können.

Überblick

Kompetenzen

Zeitaufwand	90 Minuten
Technik	touchfähige Geräte, WLAN (auch am PC mit LAN)
Methoden	Gruppenarbeit, Frage und Antwort, Simulation
Vorkenntnisse	Curriculum 1

Die Schülerinnen und Schüler ...

- ¬ festigen ihr Wissen und ihre Umsetzungskompetenz aus der letzten Stunde.
- ¬ lernen das Programmierprinzip der Schleife kennen.
- ¬ programmieren verschiedene Muster.
- ¬ wissen am Ende, wie sie es schaffen, mithilfe der Schleife einen Mäander zu zeichnen

Ablauf

Phase	Aufgabe	Methode	Zeit
Reflexion	Wiederholung des grundlegenden Wortschatzes	F & A	10‘
Arbeitsphase I	Das Sechseck	Gruppenarbeit & Simulation	25‘
	Regelmäßige n-Ecke		
	Break: Die For-Schleife		15‘
Freiarbeit	Eigene Muster	Simulation	10‘
Arbeitsphase II	Mäander	Gruppenarbeit	25‘
Rück- und Ausblick	Zusammenfassung	Plenum	5‘

2

Unterrichtsverlauf

Vorbemerkung

Die Erfahrung in den Praxiseinheiten von Code your Life zeigt, dass sich jede Klasse unterschiedlich schnell in die Thematiken einfindet. Auch innerhalb der Lerngruppen brauchen manche etwas länger und mehr Übungsphasen, andere sind ganz fix und denken bereits weiter. Auch im Handling mit dem Computer sind die Kinder unterschiedlich stark erfahren. Die Zeitangaben der vorliegenden Unterrichtseinheiten sind demnach nur ungefähre Richtwerte. Geben Sie Ihrer Lerngruppe Zeit, sowohl den Wortschatz als auch das neue Programmierprinzip der Schleife wirklich zu verstehen.

Die Kinder entwickeln nun selbstständig Ideen! Bei der Unterrichtseinheit 2 geht es nun um die sogenannten Schleifen. Mithilfe der Schleifen können Code-Abfolgen beliebig oft wiederholt werden. Das Curriculum ist so aufgebaut, dass die Kinder selbst an den Punkt kommen, sich eine Art Wiederholfunktion zu wünschen. Das Selbsterkennen hilft ungemein dabei, das Programmierprinzip zu verstehen. Unterstützen Sie Ihre Schülerinnen und Schüler dabei, Dinge selbst herauszubekommen.

Phase 1 | Reflektion

Gelerntes wiederholen! Nutzen Sie den Anfang der Stunde, um mit den Kindern die erlernten Dinge aus der letzten Unterrichtseinheit zu wiederholen. Fragen Sie die Schülerinnen und Schüler nach den grundlegenden Befehlen der Turtle. Wiederholen Sie ebenso (mündlich) die Aufgaben und Lösungswege der letzten Stunde.

Der Wortschatz noch einmal im Überblick:

♻ turtle → pen up	Die Turtle nimmt den Stift hoch.
♻ turtle → pen down	Die Turtle setzt den Stift ab.
♻ turtle → left turn (90)	Die Turtle dreht sich um 90° nach links.
♻ turtle → right turn (180)	Die Turtle dreht sich um 180° nach rechts.
♻ turtle → forward (200)	Die Turtle läuft um 200px nach vorne.
♻ turtle → back (100)	Die Turtle läuft um 100px rückwärts.
♻ turtle → set pen color (colors -> red)	Die Turtle wechselt die Farbe auf „rot“.

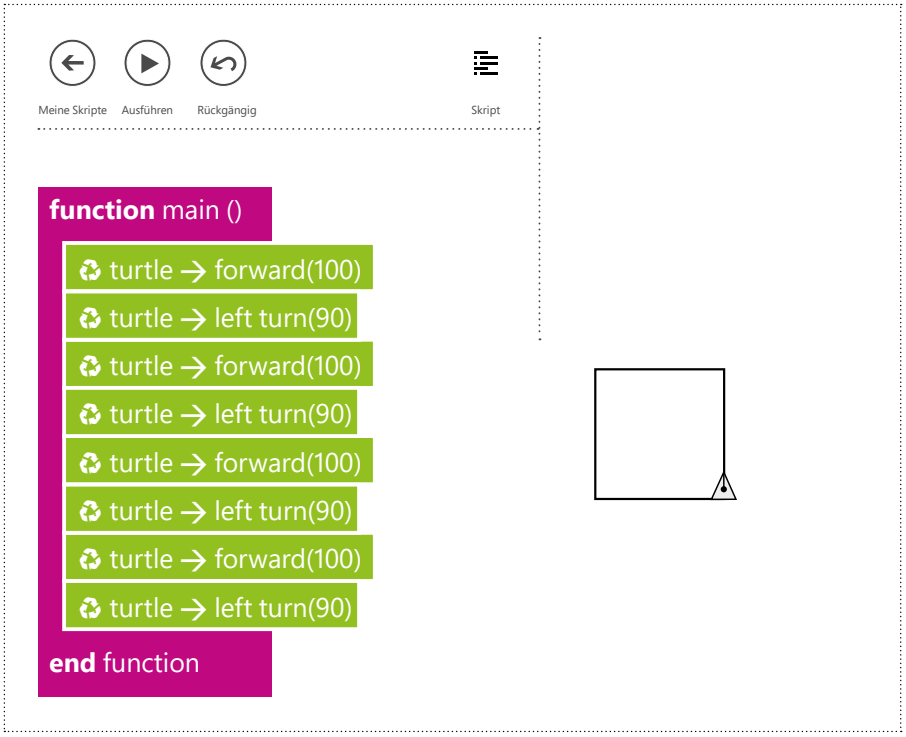
Phase 2 | Arbeitsphase I

Ziel der Arbeitsphase ist, zu erkennen, wie sich Winkel in einem regelmäßigen n-Eck berechnen lassen und wie man es schafft, dass die Turtle mit nur 2 (bzw. 4) Zeilen Code, ein beliebiges n-Eck zeichnet. Die Kinder werden in dieser Unterrichtseinheit das Programmierprinzip der Schleife kennenlernen.

2.1 Praktisches Wiederholen

Nachdem Sie die Geräte ausgeteilt haben, bzw. die Kinder die Computer hochgefahren haben, lassen Sie zuerst noch einmal ein einfaches Quadrat zeichnen.

!! Aufgabe:
Die Schildkröte soll ein Quadrat mit einer Kantenlänge von 100 Pixeln zeichnen!



CyL 2.1 Quadrat

2.2 Vorbereitung der ersten Aufgabe

Sobald die Schülerinnen und Schüler ein Quadrat auf ihrem Bildschirm haben, können Sie die nächste Aufgabe nennen.

!! Aufgabe:
Die Schildkröte soll ein regelmäßiges 6-Eck mit einer Kantenlänge von 100 Pixel zeichnen.

Zeichnen Sie ein regelmäßiges 6-Eck an die Tafel, bzw. lassen Sie einen Schüler oder eine Schülerin ein 6-Eck zeichnen. Überlegen Sie gemeinsam, was sie an den Codezeilen des Quadrats verändern müssen, um ein 6-Eck zu bekommen. Folgende Fragen bieten sich an:

- Muss man die Seitenlänge ändern? (Antwort: Nicht unbedingt.)
- Muss die Anzahl der Seiten verändert werden? Wenn ja, auf wieviel? (Antwort: Auf 6.)
- Muss der Winkel verändert werden? (Antwort: Ja)
- Und muss sich die Turtle immer gleich weit drehen? (Antwort: Ja)
- Um wieviel Grad muss sich die Turtle an jeder Ecke drehen? (Antwort: 60 Grad)

Je nach Kenntnisstand der Schülerinnen und Schüler wird es ihnen leicht oder schwer fallen, selbst auf die Winkelgröße zu kommen.

Hinweis! Helfen Sie den Schülern dabei, indem sie wieder die Schildkröte simulieren:

Stellen Sie sich vor die Schüler und laufen Sie das Quadrat ab. Zählen Sie dabei in jeder Ecke mit „1x90 Grad, 2x90 Grad, 3x90 Grad, 4x90 Grad“, bis Sie wieder an der Ausgangsposition angekommen sind. Fragen Sie die Schüler:

- Wieviel Grad habe ich mich insgesamt gedreht? (Antwort: 360 Grad)

Nun laufen Sie ein 6-Eck und zählen „1x, 2x, 3x, 4x, 5x, 6x“, bis Sie wieder an der Ausgangsposition angekommen sind. Nun wieder die Frage:

- Wieviel Grad habe ich mich jetzt insgesamt gedreht? (Antwort: Wieder 360 Grad)

Wiederholen Sie dies gegebenenfalls für ein 8-Eck. Zeigen Sie ihren Schülern so, dass sich die Turtle bei einem regelmäßigen n-Eck immer um 360 Grad dreht.

Nächste Frage:

¬ Um wieviel Grad muss ich mich bei einem Sechseck nun jedes Mal drehen?
(Antwort: 60 Grad)

Die Schüler werden vielleicht raten und z.B. 45 Grad sagen. Geben Sie die Lösung nicht vor, sondern helfen Sie den Schülern, hier selbst auf die notwendige Rechnung zu kommen. Nehmen Sie sich die Zeit und lassen Sie die Schüler nachdenken, so dass sie den Lösungsweg tatsächlich verstehen.

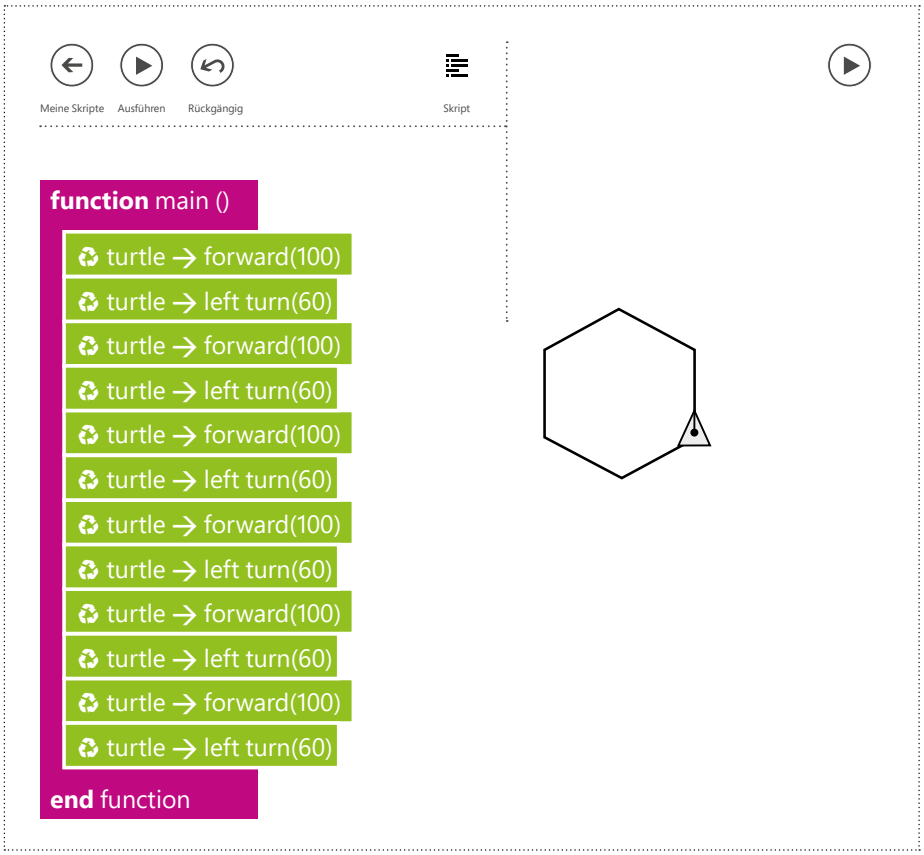
Geben Sie als Hilfestellung und für einen besseren Überblick ein Tafelbild und erstellen Sie mit den Kindern folgende Tabelle:

Anzahl Ecken	Winkelgrad der Ecken	Gradzahl gesamt
4	90	360
6	60	360
8	45	360
12	30	360
36	10	360

2.3 Das regelmäßige 6-Eck

Nun geht es an die Umsetzung des 6-Ecks auf dem Bildschirm. Geben Sie den Kindern den Hinweis, dass sie die Codezeilen des Quadrates nicht löschen müssen, sondern durch die besprochenen Veränderungen aus dem Quadratprogramm eins für 6-Ecke machen können.

!! Aufgabe: Die Schildkröte soll ein regelmäßiges 6-Eck mit einer Kantenlänge von 100 Pixel zeichnen.



CyL 2.2.3 6-Eck

2.4. Weitere n-Ecke

Je nachdem wie schnell die Teams sind, können Sie weitere Aufgaben geben:

!! Aufgabe:
Die Schildkröte soll ein regelmäßiges 8-Eck, 12-Eck, 36-Eck zeichnen.

Spätestens beim 36-Eck werden die Schülerinnen und Schüler aufstöhnen, dass das anstrengend ist. Wenn die Kinder dies selbst bemerken und bemängeln, sind sie bereit,

das erste Programmierprinzip kennenzulernen. Dies ist genau der richtige Moment, um die Schleife einzuführen.

2.5 Break - Die Schleife

Greifen Sie diese leichte Ermüdungserscheinung und Frustration der Kinder auf und fragen Sie, was Sie sich an dieser Stelle wünschen würden, z.B.

¬ Was wäre toll zu haben? (Antwort: Eine Wiederholungsschleife)

Bitten Sie zwei Schüler, z.B. Bruno und Tara nach vorne und simulieren Sie mit ihnen das Programm. Sie sind die Schildkröte. Bruno ist der Befehlsgeber, Tara der Zähler.

Brunos und Taras Aufgabe ist es nun, dass die Schildkröte, also Sie, ein 6-Eck läuft. Dazu sagt Bruno den Befehl:

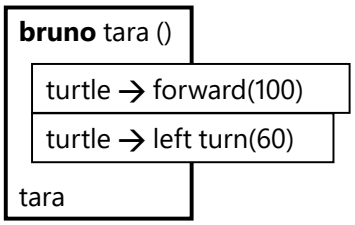
♻ turtle → forward (100)

♻ turtle → left turn (60)

Und Tara zählt diese beiden Befehle bis Sie als Turtle diese 6x durchgeführt haben. Dazu klopft sie Bruno bei jeder Zahl auf die Schulter. Wenn das 6-Eck geschafft ist, sagt Tara „Ende“.

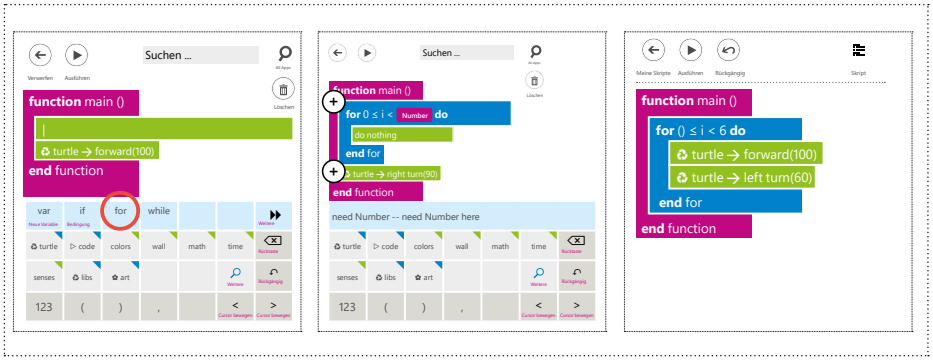
2

Malen Sie nun die Schleife folgendermaßen an die Tafel:



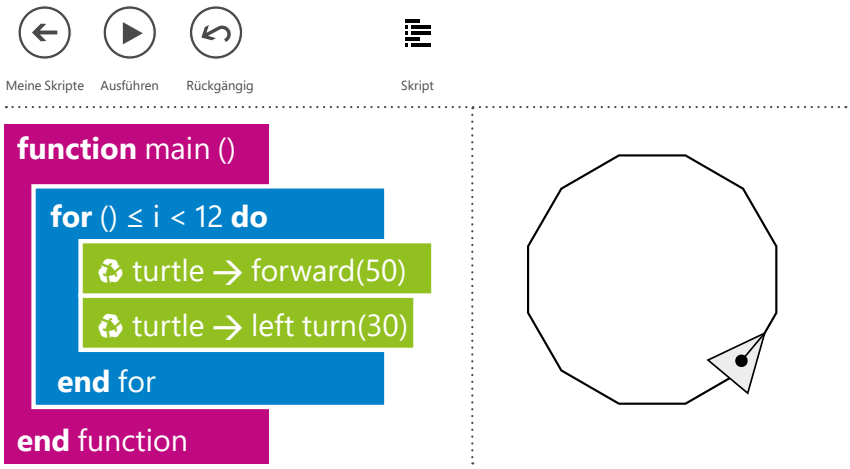
Hinweis! Wir haben gute Erfahrungen gemacht, an dieser Stelle den Namen For-Schleife etwas spielerischer zu gestalten und die Schleife nach einem Schüler zu benennen, wie in unserem Beispiel Bruno. So heißt die For-Schleife von nun an Bruno-Schleife. Sie werden sehen, dass die Kinder sich dies sehr leicht merken können und Bruno wird es stolz machen.

Erklären Sie den Kindern, dass dies nun die Bruno-Schleife ist, die in Wirklichkeit For-Schleife heißt. Lassen Sie die Schülerinnen und Schüler auf den Tastaturfeldern der Turtle nach dem passenden Befehl suchen.



Nun haben Sie das erste Programmierprinzip eingeführt. Lassen Sie nun alle Kinder die verschiedenen n-Ecke mit der Bruno-Schleife ausprobieren.

Lösungsbeispiel für ein 12-Eck



Phase 3 | Freiarbeit

3.1. Muster programmieren

Um das Schleifenprinzip zu verinnerlichen, schieben Sie eine Phase der Freiarbeit ein. So können die Schülerinnen und Schüler das neu gefundene Programmierprinzip eigenständig einüben und Routine im Umgang mit den Codezeilen bekommen. Darüber hinaus fördert die Freiarbeit die Motivation, tiefer einzusteigen. Es werden ganz verschiedene Formen und Figuren entstehen: Kreisbilder, Sterne, Zick-Zack-Linien usw.

Aufgabe:
Denkt euch verschiedene Muster aus, die ihr mithilfe der Bruno-Schleife programmieren wollt.

An dieser Stelle möchten wir Ihnen noch einen „Geheimbefehl" mit auf den Weg geben. Eventuell werden die Kinder Sie nämlich fragen, ob man die Ausführung der Turtle-Bewegungen nicht beschleunigen kann, um nicht immer so lange auf das fertige Bild warten zu müssen. Ja, kann man, mithilfe folgender Befehle:

turtle → fast	(fertiges Ergebnis wird sehr schnell angezeigt)
turtle → set speed (400)	(Geschwindigkeit wird festgelegt)

Besonders schöne Effekte erzielen die Kinder mit dem Befehl für zufällige Farbwahl:

set pen color (colors → random)

Hinweis! Je nach Aufmerksamkeits- und Aufnahmefähigkeit der Kinder können Sie die Freiarbeit zeitlich noch ausweiten. Sie können entscheiden, ob Sie die nun folgende Arbeitsphase II in die nächste Unterrichtsstunde verlagern möchten, um mit neuer Energie zu starten.

Phase 4 | Arbeitsphase II

3.2 Mäander programmieren

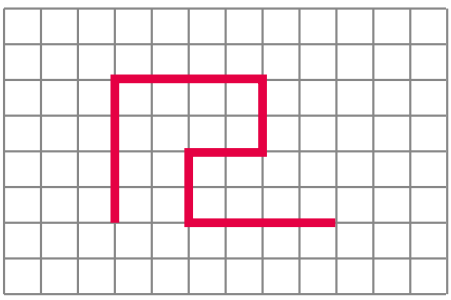
Erinnern Sie die Kinder nun an die Mäander. Stellen Sie dazu die Frage:

¬ Wozu haben wir heute die Bruno-Schleife kennengelernt? Was war die letzte Hausaufgabe? (Antwort: Mäander mitbringen.)

Lassen Sie die Schülerinnen und Schüler die Zuhause ausgedruckten Bilder herausholen. Erörtern Sie mit den Kindern, worum es sich bei einem Mäander handelt.

¬ Was ist ein Mäander? (Antwort: ein Muster, das immer wieder wiederholt wird.)

Zeichnen Sie folgenden Mäander auf die karierten Felder der Tafel und legen Sie als Hilfestellung die Längen fest: Kurze Seiten 50 Pixel, lange Seiten 100 Pixel.



Geben Sie nun die nächste Aufgabe:

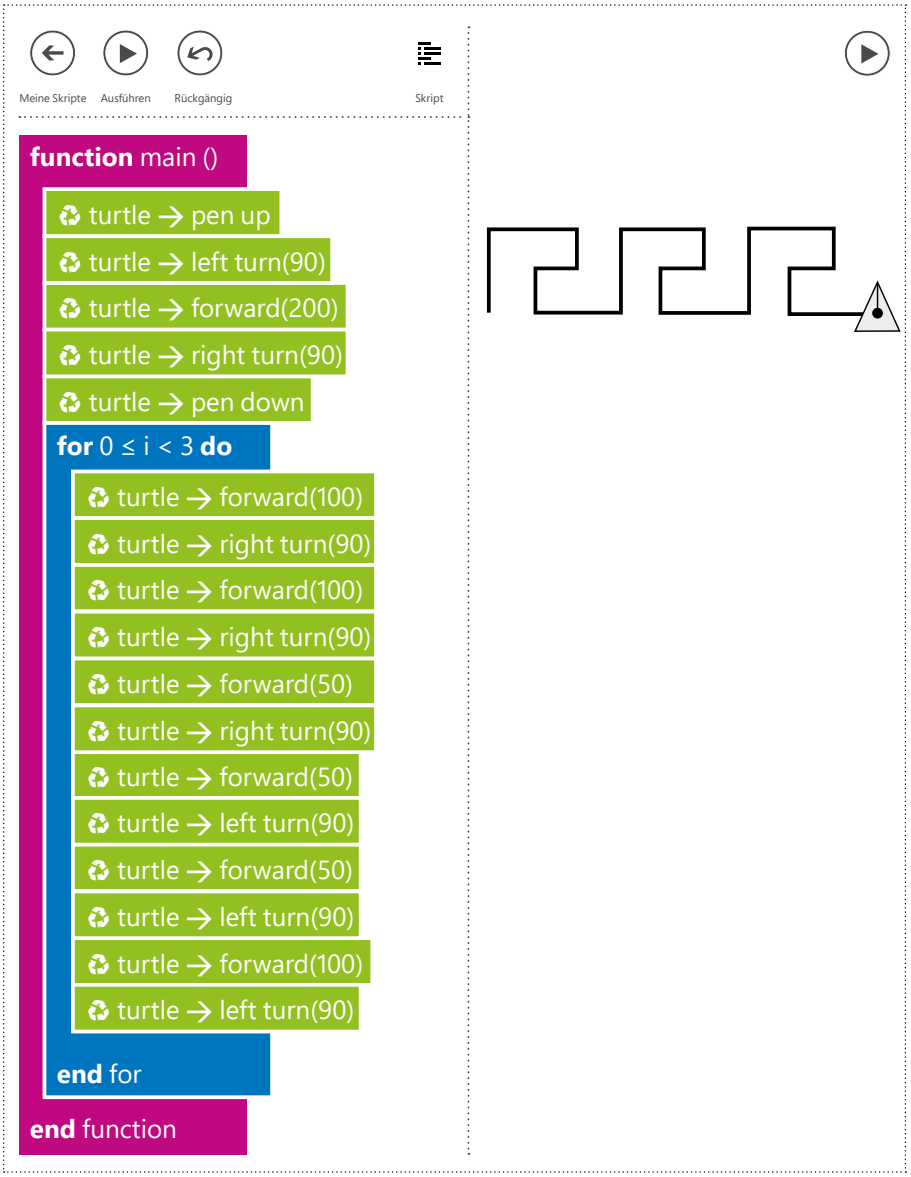
!! Aufgabe:
Lasst eure Schildkröte diesen Mäander zeichnen. Nutzt dazu die Bruno-Schleife.

Je nach Zeit, die Ihnen nun noch zur Verfügung steht, lassen Sie noch weitere Mäander in anderen Größen zeichnen.

Hinweis! Auch hier werden einige Schülerinnen und Schüler schnell auf das Ergebnis kommen, andere benötigen mehr Zeit. Lassen Sie die Kinder sich ruhig gegenseitig helfen und gehen Sie bei Schwierigkeiten die Codezeilen Schritt für Schritt durch.

2

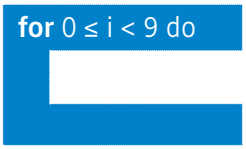
Lösungsbeispiel des Mäanders



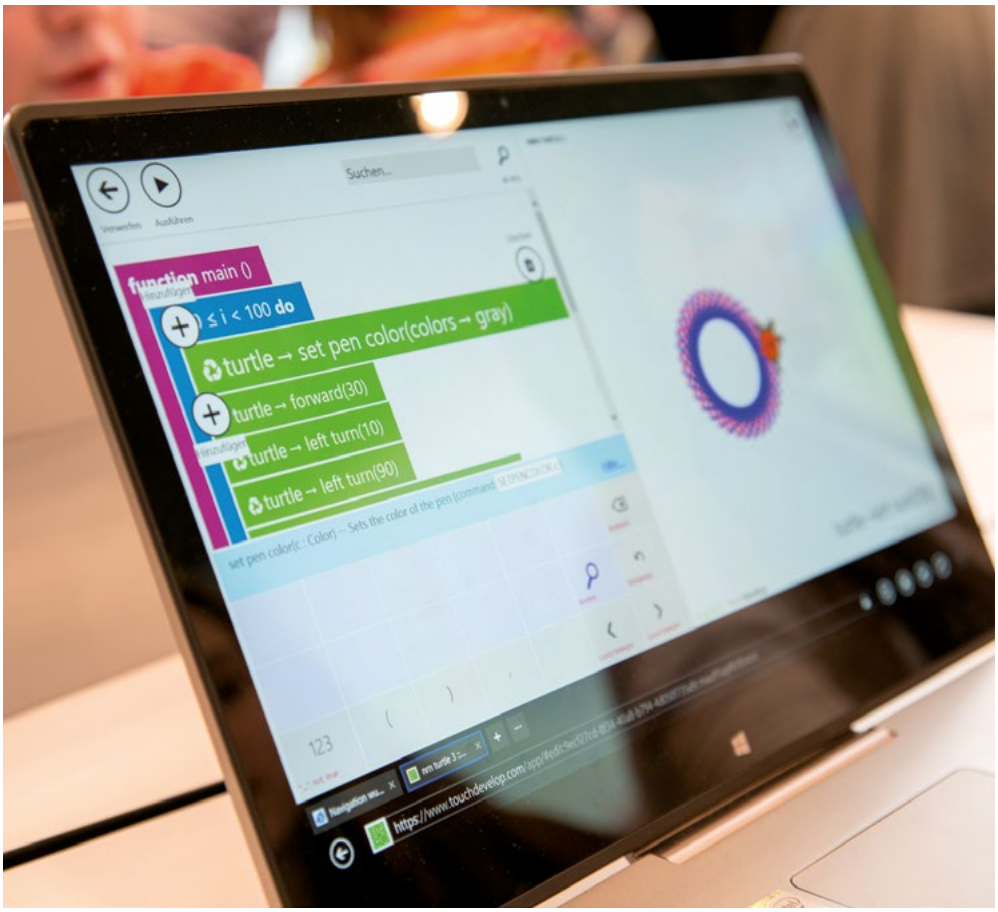
CyL 2.3.2 Mäander

Phase 4 | Ausblick

Fassen Sie am Ende noch einmal zusammen, was in dieser Unterrichtseinheit alles gemacht wurde. Geben Sie dann einen Ausblick auf das nächste Mal. Im kommenden Curriculum werden die Mäander noch weiter vertieft und ein neues Programmierprinzip wird eingeführt: Merkkästen und Variablen.

Gesamtwortschatz der Turtle (Stand Tutorial 2)	
♻ turtle → pen up	Die Turtle nimmt den Stift hoch.
♻ turtle → pen down	Die Turtle setzt den Stift ab.
♻ turtle → left turn (90)	Die Turtle dreht sich um 90° nach links.
♻ turtle → right turn (180)	Die Turtle dreht sich um 180° nach rechts.
♻ turtle → forward (200)	Die Turtle läuft um 200px nach vorne.
♻ turtle → back (100)	Die Turtle läuft um 100px rückwärts.
♻ turtle → set pen color(colors -> red)	Die Turtle wechselt die Farbe auf „rot“.
♻ turtle → fast	Die Turtle zeigt gleich das fertige Bild.
♻ turtle → set speed (400)	Die Turtle läuft in vorgegebener Geschwindigkeit.
♻ turtle → set pen color(colors->random)	Die Turtle wechselt die Farbe nach dem Zufallsprinzip.
	Mit der For-Schleife, auch Bruno-Schleife genannt, können Befehle beliebig oft wiederholt werden.

Der Nele-Merkkasten



„Ich hätte gerne eine Mäander-Kachel mit der Seitenlänge 50“, ruft der Dozent in der Rolle des Kunden ins Klassenzimmer. Fleißig verändern die Kinder den Code ihres Mäanders in die richtige Größe.

Diese dritte Unterrichtseinheit führt Variablen als neues Prinzip ein. In einer Übung simulieren die Kinder eine Fabrik, in der Kacheln mit schönen Mäandermustern hergestellt werden.

Der Dozent spielt den Kunden, der immer wieder Muster in anderen Größen verlangt.

Die Kinder kommen dabei recht schnell an einen Punkt, an dem sie erkennen, dass das Verwenden von Variablen eine große Zeitersparnis bedeutet. An den richtigen Stellen eingesetzt, hilft ihnen die Variable, den Wünschen des Kunden in Sekundenschnelle zu entsprechen.

Überblick

Zeitaufwand	90 Minuten
Technik	touchfähige Geräte, WLAN (auch am PC mit LAN)
Methoden	Gruppenarbeit, Frage und Antwort, Simulation
Vorkenntnisse	Curriculum 1 und 2

Kompetenzen

Die Schülerinnen und Schüler ...

- ¬ festigen ihr Wissen und ihre Umsetzungskompetenz aus der letzten Stunde (For-Schleife).
- ¬ lernen das Verwenden von Variablen beim Programmieren kennen.
- ¬ wissen am Ende, wie sie es schaffen, mithilfe von Schleifen und Variablen Mäander mit nur wenigen Veränderungen im Code zu zeichnen

Ablauf

Phase	Aufgabe	Methode	Zeit
Reflexion	Wiederholung des grundlegenden Wortschatzes und der For-Schleife	F & A	25'
Arbeitsphase	Idee: Die Mäander-Fabrik	Gruppenarbeit & Simulation	45'
	Mäander mit Bruno und Nele		
Freiarbeit	Eigene Mäander entwerfen	Gruppenarbeit	15'
Ausblick		Hausaufgabe	5'

3

Unterrichtsverlauf

Vorbemerkung

Bei der Unterrichtseinheit 3 geht es um den Einsatz von Variablen. Diese ermöglichen, Elemente des Codes mit nur einer Änderung zu variieren. Entsprechend des Curriculums kommen die Kinder an einen Punkt, an dem sie erkennen, dass das Verwenden von Variablen eine große Zeitersparnis und viele neue Möglichkeiten bedeutet. Das Selbsterkennen hilft ungemein dabei, das Programmierprinzip zu verstehen. Unterstützen Sie Ihre Schülerinnen und Schüler dahingehend dabei, Dinge selbst herauszubekommen.

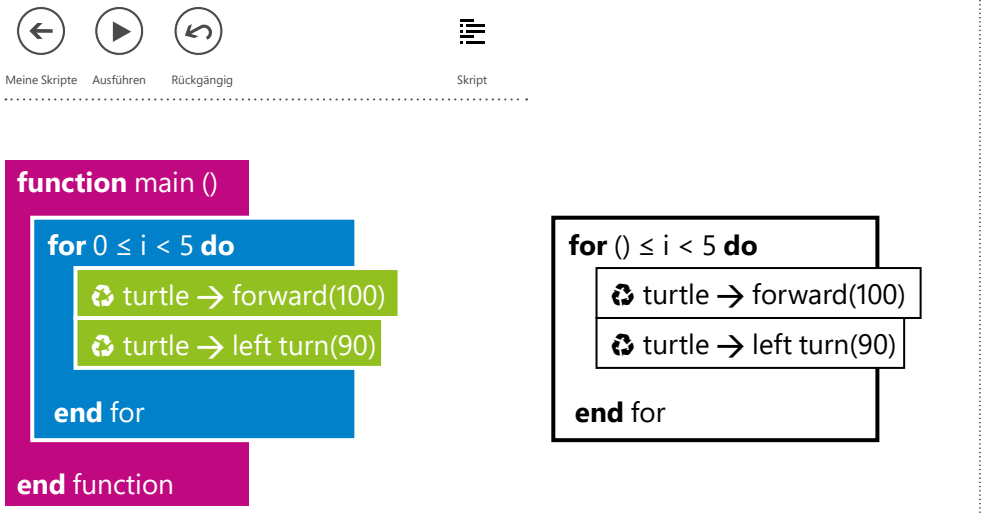
Das Curriculum zu den Variablen ist für 90 Minuten konzipiert. Je nach Alter und Lerngruppenzusammensetzung können Sie sich gerne auch zwei Unterrichtsblöcke Zeit lassen. Dies gibt den Schülerinnen und Schülern die Möglichkeit, die Problemstellungen und Aufgaben selbst zu lösen und eine gewisse Routine im Umgang mit dem bisher gelernten Wortschatz, dem Einsatz der Bruno-Schleife und der Verwendung von Variablen zu erlangen.

Phase 1 | Reflexion

Gelerntes wiederholen! Nutzen Sie den Anfang der Stunde, um mit den Kindern die erlernten Dinge aus den letzten beiden Unterrichtseinheiten zu wiederholen. Sie werden feststellen, dass die grundlegenden Befehle aus Curriculum 1 ohne Probleme wiedergegeben werden können. Fragen Sie die Schülerinnen und Schüler nach der Aufgabe der letzten Stunde und mit welchem Programmierprinzip sich diese einfach und schnell umsetzen ließ: das Zeichnen eines Mäanders mit Hilfe der Bruno-Schleife.

Die Kinder werden sich leicht an den Namen Bruno-Schleife erinnern. Bitten Sie die Kinder nun für die Simulation der letzten Stunde nach vorn und simulieren Sie mit ihnen gemeinsam ein 8-Eck. Rufen Sie hier den Kindern noch einmal den richtigen Namen (For-Schleife) ins Gedächtnis, da dieser für das Eingeben des Codes entscheidend ist.

Wiederholen Sie gemeinsam mit den Kindern, wie die Bruno-Schleife in der Turtle-Sprache aussah. Für die bildhafte Vorstellung bietet es sich an, an der Tafel mit den gleichen Farben zu arbeiten, die auch im Programm zu sehen sind, d.h. die Bruno-Schleife in blauer Farbe, als Klammer um die grünen Befehle.



Beispiel einer FOR-Schleife im Programm **Beispiel einer FOR-Schleife an der Tafel**

Bei der Wiederholung bietet es sich an, mit den Schülerinnen und Schülern noch einmal genauer die Ziffern und Symbole rund um den „Zähler" i zu betrachten.

Hinweis! Zeichnen Sie zur Erläuterung eine Tabelle an die Tafel und probieren Sie gemeinsam mit den Schülerinnen und Schülern, was < im Gegensatz zu ≤ bedeutet und wie sich die Zahlen der Bruno-Schleife ergeben.

Die Tabelle enthält dabei zwei Spalten:

1. i als Variable für den Zähler
2. Abzählungen, d.h. wie häufig tippt das Kind, welches den Zähler simuliert, auf die Schulter des anderen Kindes

Am Beispiel For 0 < i < 5 bedeutet dies:

i	Abzählungen
0	1
1	2
2	3
3	4
4	5

Damit erklärt sich den Schülerinnen und Schülern, wieso die Befehle der For-Schleife bereits das erste Mal ausgeführt werden, obwohl als Ziffer des Zählers eine 0 steht ($0 \leq i$) und wieso dennoch der Zähler, wie in unserem Beispiel, bei der Zählerzahl 4 endet ($i < 5$), trotzdem aber 5 Wiederholungen der grün unterlegten Befehle durchgeführt werden.

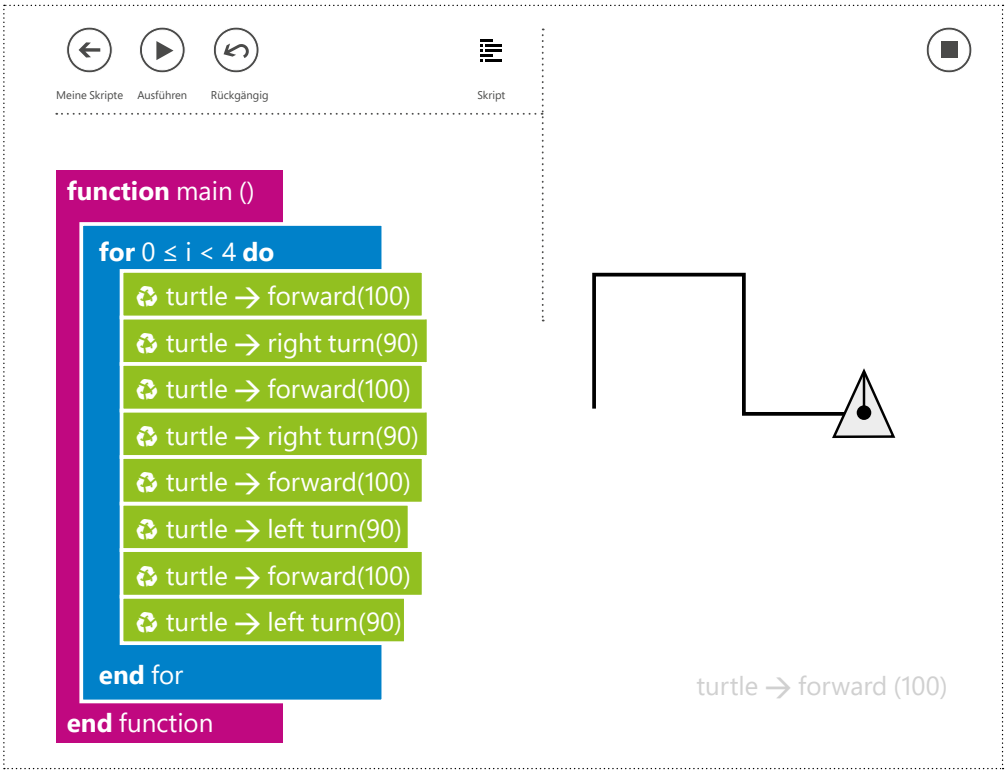
Hinweis! Rufen Sie an dieser Stelle noch einmal den Nutzen der Bruno-Schleife in Erinnerung: sie erspart jede Menge Arbeit und Zeit. Als Bespiel erinnern Sie gern an die Gestaltung eines 36-Eck und das mühselige Eingeben von 72 immer gleichen Befehlen.

Bitten Sie eines der Kinder an die Tafel zu kommen und den Mäander der letzten Stunde anzuzeichnen. Lenken Sie dabei den Blick auf die Blickrichtung der Turtle nach Durchlauf der Befehle in der Bruno-Schleife. Wichtig ist an dieser Stelle der Hinweis, dass als letzter Befehl eine Drehung notwendig ist, so dass die Turtle, ebenso wie zu Beginn, nach oben schaut.

Nachdem Sie die Geräte ausgeteilt haben bzw. die Kinder die Computer hochgefahren haben, lassen Sie die Kinder ein einfaches Mäander mit Hilfe der Turtle zeichnen, z.B. wie nachfolgend gezeigt.

!! Aufgabe:
Die Schildkröte soll ein vorgegebenes Mäander-Muster mit eine Kantenlänge von 100 Pixel zeichnen.

Lösungsbeispiel



CyL 3.1.1 Mäander

Die Praxiseinheiten von Code your Life in Berliner Schulen hat gezeigt, dass die Kinder oftmals ganz selbstverständlich zu Beginn den PEN UP-Befehl nutzen und ihre Turtle an den Rand des Bildschirms laufen lassen, um alle Mäander-Wiederholungen zu sehen.

Hinweis! Sollten Kinder Probleme bei der Darstellung der Mäander haben, ermutigen Sie sie, den Mäander im Raum abzulaufen. Es hilft z.B. bei der Entscheidung, in welche Richtung sich die Turtle drehen muss.

Phase 2 | Arbeitsphase

Ziel der Arbeitsphase ist, zu erkennen, wie sich durch wenige Änderungen im Code vorgefertigte Mäander-Muster verändern lassen. Die Kinder werden in dieser Unterrichtseinheit lernen, wie praktisch variablen sind.

2.1 Idee: die Mäander-Fabrik

Entwickeln Sie mit den Schülerinnen und Schülern die Idee, eine Mäander-Fabrik zu gründen. Stellen Sie sich gemeinsam vor, dass ein Kunde zur Tür herein kommt und sich ein konkretes Mäander-Muster in x-facher Ausführung und mit einer bestimmten Seitenlänge wünscht. Öffnen Sie dafür z.B. spielerisch die Tür des Klassenraums und spielen Sie den Kunden:

„Ich hätte gern ein Mäander mit einer Seitenlänge von 50 Pixel in 7facher Ausführung." Werfen Sie die Frage auf, an wie vielen Stellen eine Ziffer geändert werden müsste, um diesen Auftrag umzusetzen.

Im oben gezeichneten Beispiel an fünf Stellen:
- einmal muss die Seitenlänge des Mäanders an vier Stellen verändert werden
- und die Anzahl der Wiederholungen muss in der Bruno-Schleife einmal angepasst werden.

Machen Sie den Kindern klar, dass dieses Beispiel ein einfaches Muster eines Mäanders darstellt. Die Zahl der Änderungen könnte auch deutlich höher sein. Ein Kunde hat allerdings nicht so viel Zeit, so lange zu warten bzw. ist die Schlange der Kunden sehr lang, sie wollen schnell ein Ergebnis sehen.

Sammeln Sie gemeinsam mit den Kindern Ideen, wie sich das Problem lösen ließe.
Die Kinder werden Ideen nennen wie, z.B. verschiedene Muster vorab anzulegen und dann entsprechend auf die Kundenwünsche reagieren zu können durch die Abänderung der Seitenlänge oder dass man alle Stellen, die geändert werden müssten, markiert werden und nur mit einem Klick die Ziffern wechseln.

Bitten Sie drei Kinder nach vorne, z.B. Nele, Maria und Paul. Paul darf die Schildkröte sein und den einfachen Mäander ohne Wiederholung laufen. Maria ist das Programm, das auf Instruktionen wartet. Und Nele bekommt vom Kunden die wichtigen Informationen. Flüstern Sie nun Nele die gewünschte Seitenlänge ins Ohr. Nele ist nun der Merkkasten des Programms und flüstert Maria bei jedem Forward-Befehl zu, wie weit Paul, die Schildkröte, vorwärts gehen soll. Und Paul läuft den gewünschten Mäander.

Flüstern Sie Nele unterschiedliche Zahlen zu und erhöhen Sie gern die Geschwindigkeit.

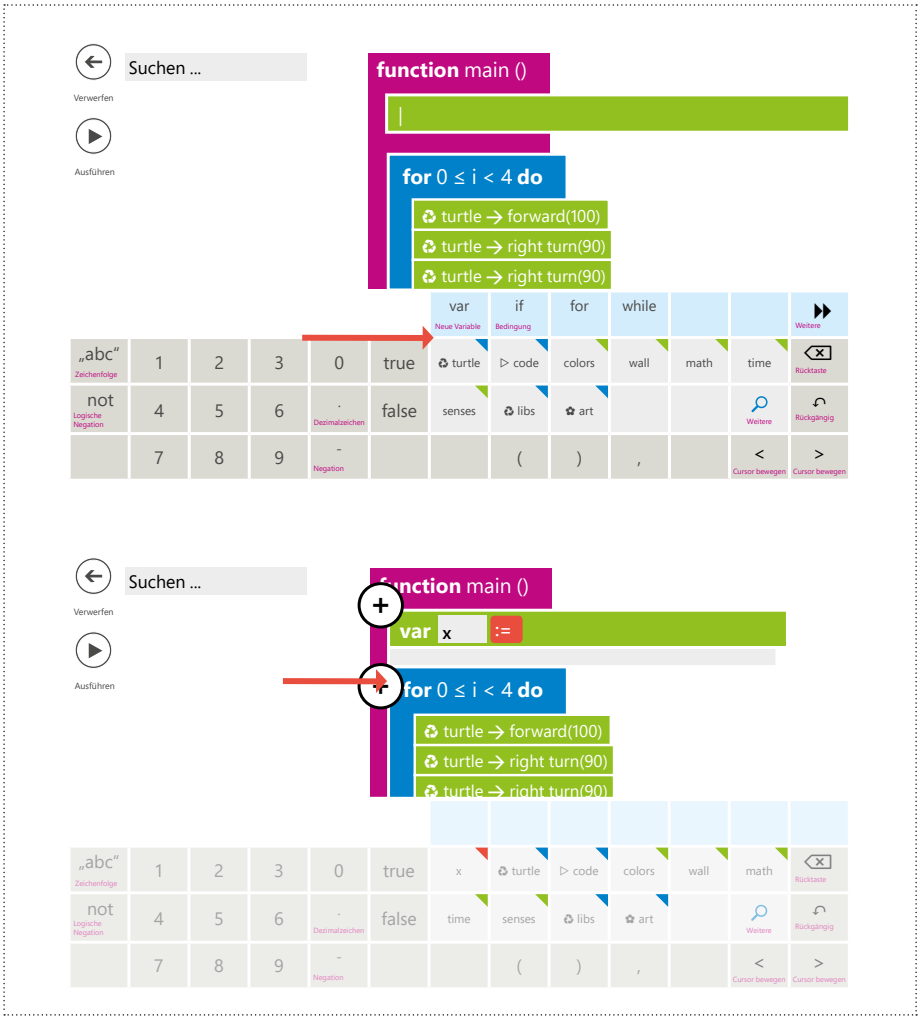
Nach dieser kleinen Simulation haben die Kinder ein weiteres Programmierprinzip kennengelernt, nämlich die Variablen.

Weil dies nun wieder ein sperriger Begriff ist, verwenden wir auch hier wieder den Namen eines Schülers. In unserem Beispiel ist das Nele. Das Prinzip der Variablen heißt von nun an Nele-Merkkasten.

Dies ist der richtige Zeitpunkt, um den Nele-Merkkasten am Tablet auszuprobieren. Lassen Sie die Kinder nach einem Befehl suchen, der zu Variablen passt.

Hinweis! Sie finden die Variablen über die Schaltfläche „var" im hellblauen Bereich des Menüfeldes. In diesem Merkkasten stecken die Wünsche des Kunden.

Fragen Sie die Kinder, an welche Stelle im Programm sie die Variable setzen würden. Schnell werden die Schülerinnen und Schüler erkennen, dass diese über der Bruno-Schleife stehen muss.



In der Code-Folge finden Sie dann die Zeile

var x := ...

x symbolisiert dabei die Variable, hinter dem := steht die Zahl, die „zugeflüstert" wurde, z.B. die Seitenlänge des Mäanders.

Hinweis! Nutzen Sie die Gelegenheit und erklären Sie den Kindern, dass eine Variable, sprich der Nele-Merkkasten, nicht immer **„x"** heißen muss. Beim Programmieren nutzt man idealerweise auch Worte als sogenannte sprechende Nele-Merkkästen.

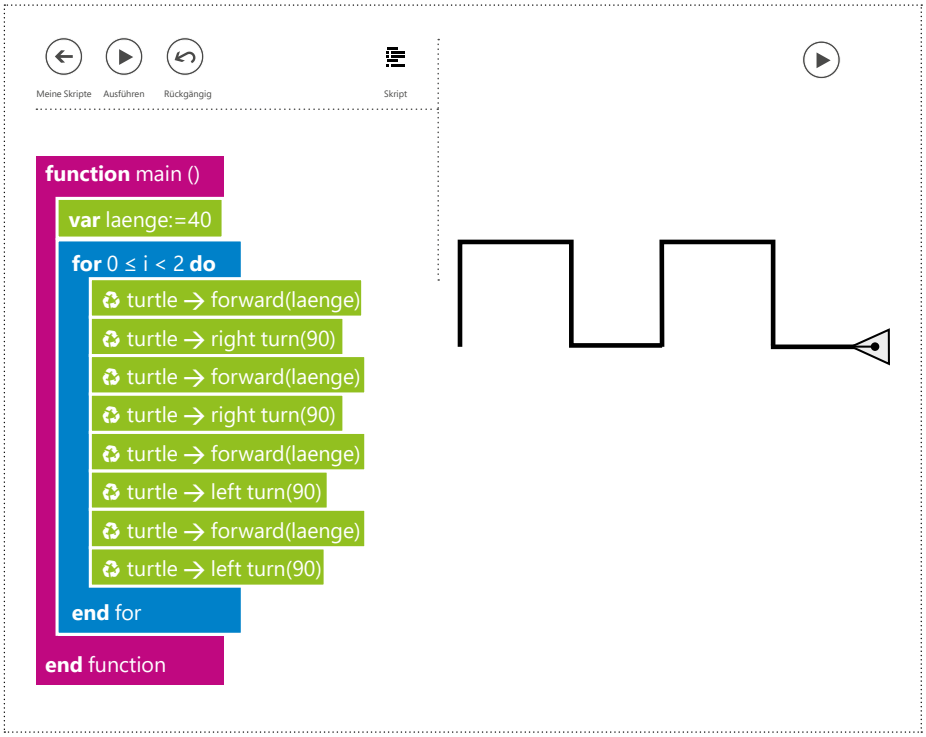
Im vorliegenden Fall wäre das zum Beispiel „laenge“, weil ja die Seitenlänge verändertwerden soll. So sehen bzw. lesen wir auf den ersten Blick, was sich der Nele-Merkkasten merkt. Bitten Sie die Kinder nun, statt des „x“ die Variable „laenge“ einzugeben.

Fordern Sie die Kinder auf, den Nele-Merkkasten „laenge“ entsprechend in die Code-Folge einzuarbeiten, an allen Stellen, wo sie die Streckenlänge des Mäanders eingaben.

!! Aufgabe:

Ersetzt die Seitenlänge eures Mäanders durch die Variable „laenge“. Die Schildkröte soll dann das vorgegebene Mäander-Muster mit eine Kantenlänge von 40 Pixel zeichnen.

Lösungsbeispiel



CyL 3.2.1 Mäander

An jede Stelle im Code, wo die Variable „laenge“ eingesetzt wurde, wird dieses bei der Ausführung des Programms automatisch durch den benannten Wert, in unserem Fall den Wert 40, ersetzt. Käme jetzt ein Kunde in unsere Mäander-Fabrik, brauchen wir

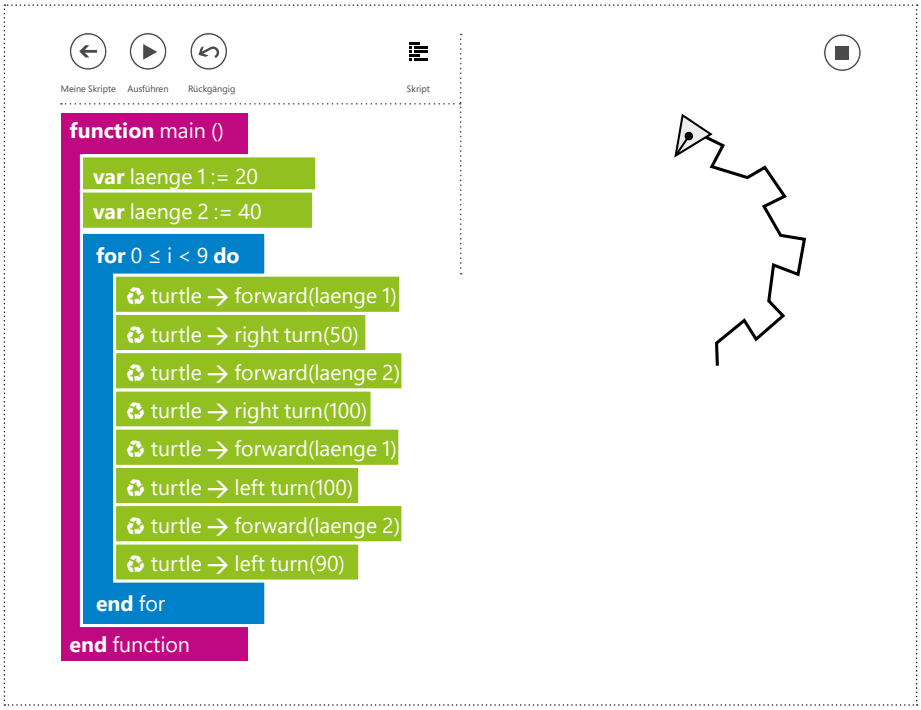
nur die Ziffer des Nele-Merkkastens ändern und sofort erhielte er das voreingestellte Mäander-Muster.

Phase 3 | Freiarbeit

Die Phase der Freiarbeit dient dazu, dass die Schülerinnen und Schüler nun die Möglichkeit haben, das neu gefundene Programmierprinzip einzuüben und Routine im Umgang mit den Codezeilen zu bekommen. Darüber hinaus fördert die Freiarbeit die Motivation.

3.1 Eigenes Mäander

Die Schülerinnen und Schüler haben in der Phase der Freiarbeit die Gelegenheit, eigene Mäander-Muster zu entwerfen, wobei die Seitenlänge nicht immer einheitlich sein muss, auch die Gradzahl der Winkel kann variieren. So gestaltet jede Gruppe ein anderes Mäander-Muster, immer unter Verwendung der Variablen (natürlich können auch mehrere Variablen genutzt werden).



CyL 3.3.1 Mäander

Phase 4 | Ausblick

Haben Sie am Ende der Unterrichtseinheit noch Zeit, lassen Sie die Kinder noch ein wenig frei „spielen“ und Befehle für die Turtle ausprobieren.

In der nächsten Unterrichtseinheit wird weiter in der Mäander-Fabrik gearbeitet und das Programmierprinzip der Variable des Nele-Merkkasten in Kombination mit weiteren Operatoren eingeübt.

Gesamtwortschatz der Turtle (Stand Tutorial 3)	
♻ turtle → pen up	Die Turtle nimmt den Stift hoch.
♻ turtle → pen down	Die Turtle setzt den Stift ab.
♻ turtle → left turn (90)	Die Turtle dreht sich um 90° nach links.
♻ turtle → right turn (180)	Die Turtle dreht sich um 180° nach rechts.
♻ turtle → forward (200)	Die Turtle läuft um 200px nach vorne.
♻ turtle → back (100)	Die Turtle läuft um 100px rückwärts.
♻ turtle → set pen color (colors -> red)	Die Turtle wechselt die Farbe auf „rot“.
♻ turtle → set pen color (colors->random)	Die Turtle wechselt die Farbe nach dem Zufallsprinzip.
♻ turtle → fast	Die Turtle zeigt gleich das fertige Bild.
♻ turtle → set speed (400)	Die Turtle läuft in vorgegebener Geschwindigkeit.
var laenge := 50	Im Nele-Merkkasten (Variable) laenge wird ein Wert (50) gemerkt.
for 0 ≤ i < 9 do	Mit der For-Schleife, auch Bruno-Schleife genannt, können Befehle beliebig oft wiederholt werden.

Thu Thao ist schlau

Die vierte Unterrichtseinheit führt in die Möglichkeit ein, Rechenvorgänge mit Variablen zu kombinieren, um komplexe Skripte vereinfacht darstellen zu können.

Spätestens jetzt zeigt sich die enge Verknüpfung von Programmieren und Mathematik. Das logische Denken und Abstraktionsvermögen der Schülerinnen und Schüler wird herausgefordert und findet in einem für sie neuen Anwendungsfeld Platz.

"Was, das ist Mathe? Dann ist Mathe ab heute mein Lieblingsfach." Diese schöne Rückmeldung einer Schülerin der Pilotphase ist Sinnbild für die Bedeutung anwendungsbezogenen Lernens und zeigt, wie begeistungsfähig die Schülerinnen und Schüler in dieser Altersgruppe für logisches Denken und Programmieren sind.

Überblick

Zeitaufwand	90 Minuten
Technik	touchfähige Geräte, WLAN (auch am PC mit LAN))
Methoden	Gruppenarbeit, Frage und Antwort, Simulation
Vorkenntnisse	Curriculum 1 bis 3

Kompetenzen

Die Schülerinnen und Schüler ...

- ¬ festigen ihr Wissen und ihre Umsetzungskompetenz aus den letzten Stunden, indem sie sowohl die Bruno-Schleife, als auch den Nele-Merkkasten immer wieder einsetzen und verwenden.
- ¬ lernen, dass Sie ein Computerprogramm auch nutzen können, um Rechenvorgänge zu lösen.
- ¬ wissen am Ende, wie sie es schaffen, mithilfe von Rechenvorgängen Strukturen zu vereinfachen.

Ablauf

Phase	Aufgabe	Methode	Zeit
Reflexion	Wiederholung der letzten Stunde	F & A	15‘
Arbeitsphase	Das Winkelband	Gruppenarbeit & Plenum	70‘
	Der komplizierte Mäander		
Ausblick	Die Spirale	Hausaufgabe	5‘

Unterrichtsverlauf

Vorbemerkung

Die Unterrichtseinheit 4 vertieft den Einsatz von Variablen und zeigt auf, wie mit den Variablen auch gerechnet werden kann.

Achten Sie darauf, welchen Kenntnisstand im Dividieren und Multiplizieren ihre Lerngruppe hat. Die Unterrichtseinheit ist recht niedrigschwellig angelegt und arbeitet nicht mit komplizierten Ausdrücken. So wird nur mit ganzen Zahlen addiert, multipliziert und dividiert. Sie können jedoch die Aufgabenstellungen an dieser Stelle beliebig erweitern, vereinfachen oder erschweren und andere Rechenvorgänge einbauen, wenn die Schülerinnen und Schüler bereits über ein ausreichendes Wissen verfügen. Die Herausforderungen sollten an die Lerngruppe angepasst werden.

Phase 1 | Reflexion

Stellen Sie auch zu Beginn dieser Stunde eine kleine Reflektion voran und greifen Sie das Thema der letzten Stunde wieder auf. Rufen Sie die Mäanderfabrik ins Gedächtnis und lassen Sie die Kinder noch einmal ohne Rechner die Situation simulieren. Die Kinder haben sich bestimmt den Namen Nele-Merkkasten gemerkt.

Fragen Sie die Kinder, was das besondere an der Fabrik war. Gehen Sie darauf ein, dass mithilfe des Merkkastens verschiedene Muster recht schnell in ihrer Größe variiert werden können. Dokumentieren Sie gemeinsam mit den Kindern an der Tafel, wie das Skript des Mäanders inklusive Bruno-Schleife und Nele-Merkkasten aussah. Idealerweise verwenden Sie auch hier die passenden Farben aus der Programmierumgebung.

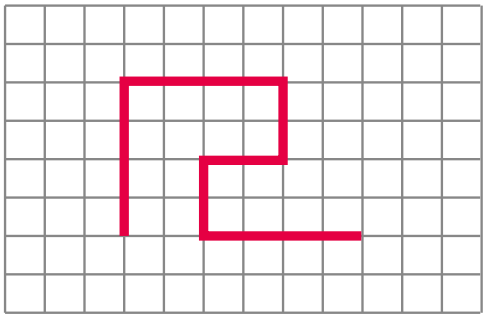
Phase 2 | Arbeistphase

Ziel der Arbeitsphase ist es, zu erkennen, dass uns das Computerprogramm Berechnungsvorgänge abnehmen kann und so selbst komplizierte Mäander mit verschiedenen Seitenlängen mit nur einem Nele-Merkkasten dargestellt werden können.

2.1 Das Winkelband

Wir kommen an dieser Stelle auf den Mäander zurück, der in Curriculum 2 eine Rolle spielte.

Zeichnen Sie den Mäander noch einmal an die Tafel, idealerweise auf die karierten Felder. Geben Sie als Hilfestellung die Länge der Seiten an, z.B. lange Seite 200 Pixel, kurze Seite 100 Pixel.



Stellen Sie nun die Aufgabe:

!! Aufgabe:
Lasst die Turtle den Mäander des Tafelbildes zeichnen, so dass ein langes sogenanntes Winkelband entsteht. Nutzt dabei die Bruno-Schleife und zwei Merkkästen.

Fordern Sie nun nach und nach die Kleingruppen auf, die Größe des Mäanders zu ändern. Z.B. „Macht den Mäander mal größer in 400/200 Pixel Seitenlänge. Oder kleiner in 100/50 Seitenlänge".

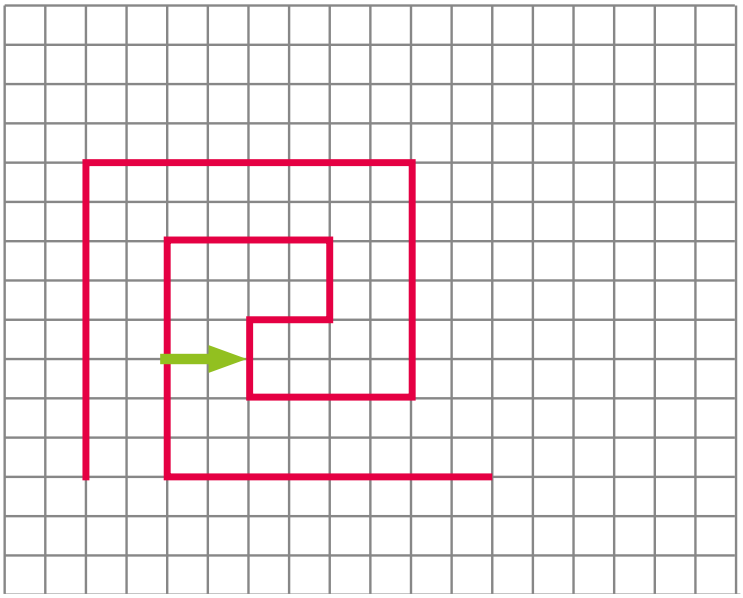
Variation: Für die schnelleren Schülerinnen und Schüler können Sie eine kleine Herausforderung einbauen: Geben Sie die Aufgabe, den Mäander mit der langen Seitenlänge 150 Pixel zu programmieren. Die kurze Seitenlänge verraten Sie vorerst nicht. Hier müssen die Kinder erst einmal überlegen. (Antwort 75 Pixel)

Haben alle Schülerinnen und Schüler den Mäander in mindestens zwei unterschiedlichen Größen ausgeführt, lenken Sie die Aufmerksamkeit wieder nach vorne und bereiten Sie die nächste Aufgabe vor.

2.2 Der komplizierte Mäander

Nach dieser Vorübung werden wir uns nun einem komplizierterem Muster widmen, um nach und nach zu zeigen, dass Computerprogramme rechnen können.

Zeichnen Sie folgenden Mäander an die Tafel.



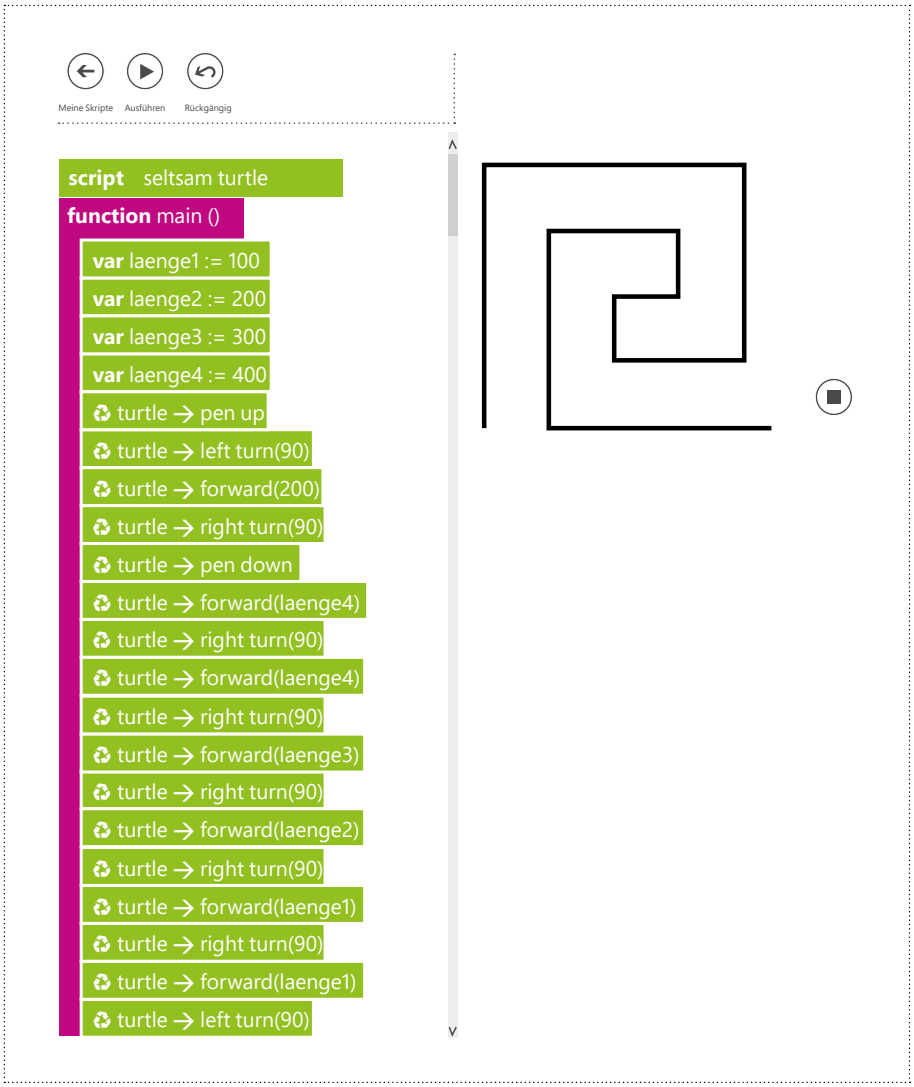
Bevor die Kinder loslegen und den Mäander programmieren, überlegen Sie ein paar Fragen gemeinsam mit allen Kindern:

¬ Wie viele Merkkästen braucht ihr nun? Wieder nur zwei? (Antwort: Nein, 4)

Markieren Sie die Seiten und benennen Sie die jeweiligen Seitenlängen für den Nele-Merkkasten durch: laenge1, laenge2, laenge3, laenge4. Beginnen Sie mit lanege1 an der kürzesten Seite. (grüner Pfeil)

Nun können Sie die Aufgabe stellen und die Kinder erst einmal loslegen lassen.

!! Aufgabe:
Die Turtle soll das neue Muster zeichnen. Die kleinste Seite beträgt 100 Pixel.



CyL 4.2.2 Mäander

Hinweis! Dieser Mäander braucht etwas Zeit und die Kinder müssen buchstäblich etwas um die Ecke denken. Lassen Sie den Kindern genug Zeit und ermuntern Sie die Schnelleren, ihren Klassenkameraden zu helfen.

Sind die Ersten bereits fertig, lassen Sie die Kinder nun wieder die Größe ihres Mäanders ändern.

!! Aufgabe:
Die Turtle soll das Muster noch einmal zeichnen.
Nun soll die kleinste Seite 50 Pixel betragen.

(Variante für die Schnelleren: Die kleinste Seite soll 25 Pixel betragen)

¬ Lassen Sie die Kinder die Aufgabe durchführen und reflektieren Sie gemeinsam mit den Kindern, an wieviel Stellen sie etwas ändern mussten. (Antwort: an 4)

¬ Fragen Sie die Kinder: Geht das auch mit weniger Veränderungen?
Schauen Sie sich um, bestimmt haben bei dieser Aufgabe einige Schüler einen Zettel und Stift geschnappt und haben schriftlich ausgerechnet, was an jeder Seitenlänge stehen soll.

Diese Schüler sind schlau. So wie Thu Thao, eine Schülerin aus der Pilotphase. Sie hat für die Gruppe immer auf dem Papier addiert, wie lang die Seiten sind. Sie hat also gerechnet. Thu Thao ist also schlau. Doch auch das Computerprogramm ist schlau. Es kann auch rechnen.

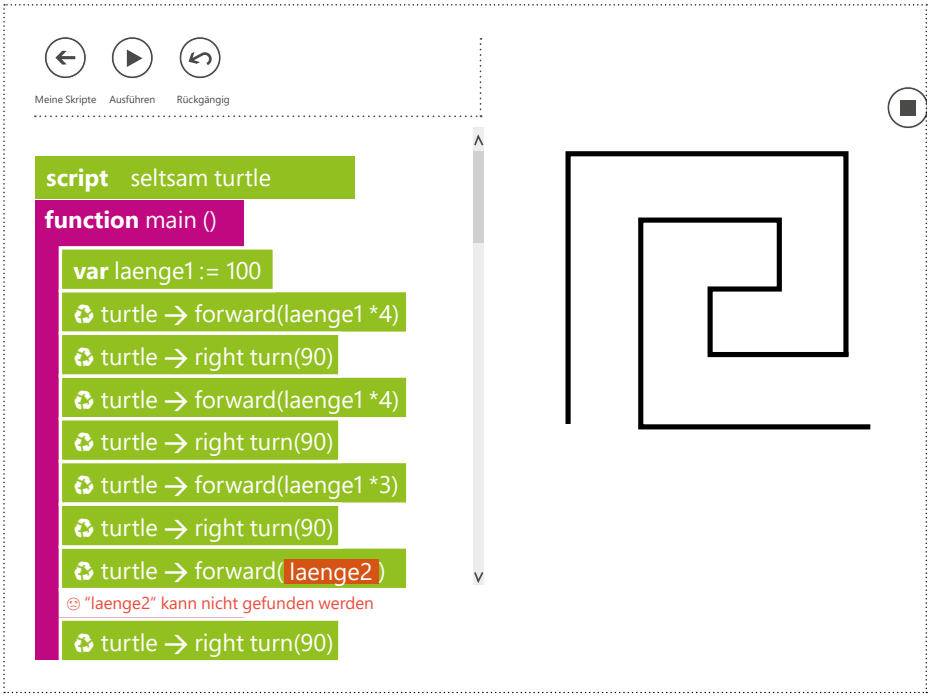
¬ Fragen Sie die Kinder: Was habt ihr gerechnet?

Überlegen Sie mit den Kindern, ausgehend von der kleinsten Seite, was jeweils gerechnet werden muss. Dokumentieren Sie dies auf der Tafel mit einer Tabelle.

	Runde 1	Runde 2	Runde 3	Rechenvorgang
laenge1	100	50	25	laenge1
laenge2	200	100	50	2*laenge1
laenge3	300	150	75	3*laenge1
laenge4	400	200	100	4*laenge1

Lassen Sie die Kinder nun die Merkkästen austauschen.
¬ Wie viele Merkkästen brauchen wir nun nur noch? (Antwort: Einen)

Das Skript sieht nun am Anfang folgendermaßen aus: (Den Gang der Turtle zur unteren Ecke ist in diesem Beispiel der Übersichtlichkeit halber ausgespart)



TouchDevelop gibt Hinweise, wenn sich im Skript ein Fehler befindet. Im obigen Beispiel steckt in einem Parameter noch der alte Nele-Merkkasten. Da dieser aber nicht definiert wurde, kennt das Programm ihn nicht und gibt eshalb die Fehlermeldung aus.

Die Kinder werden in unterschiedlichem Tempo zu diesem Mäander kommen. Geben Sie ihrer Lerngruppe die Chance, das eben Gelernte zu verinnerlichen und lassen Sie die Kinder noch ein wenig mit dem Thu Thao Prinzip experimentieren und die Größen variieren. Greifen Sie gerne die Simulation der Fabrik noch einmal auf. Spannend wird es, wenn die Kinder den Mäander mit der Bruno-Schleife in lange Bänder verwandeln.

Starten Sie einen kleinen Wettbewerb:

!! Aufgabe:
Wer schafft den längsten Mäander?

Die Kinder werden versuchen, die Seitenlängen möglichst klein zu machen, um möglichst viele Wiederholungen programmieren zu können. Vielleicht schafft es sogar jemand den ganzen Bildschirm mit dem Mäander zu füllen?

Lösungsbeispiel:

Meine Skripte Ausführen Rückgängig

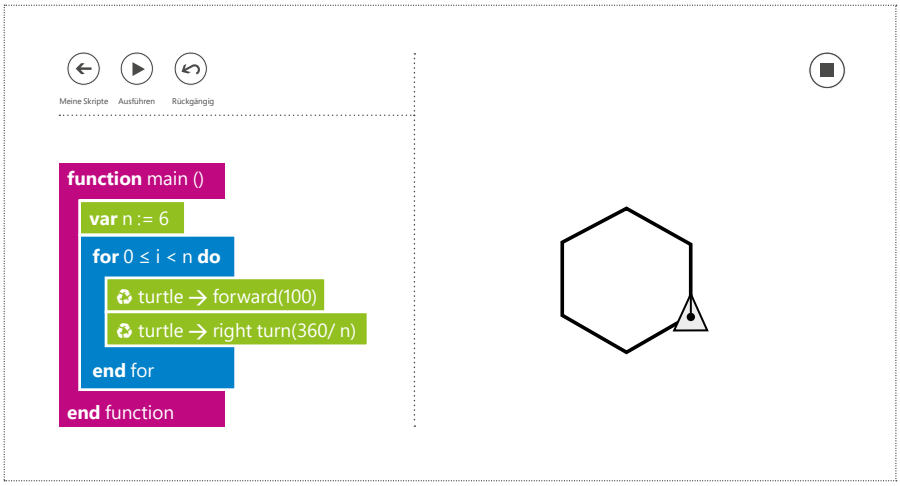
```
script  wunderbar turtle (2)
function main ()
    turtle → left turn(90)
    turtle → pen up
    turtle → forward(200)
    turtle → right turn(90)
    turtle → pen down
    for 0 ≤ i < 20 do
        var laenge :=3
        turtle → forward(laenge * 4)
        turtle → right turn(90)
        turtle → forward(laenge * 4)
        turtle → right turn(90)
        turtle → forward(laenge * 3)
        turtle → right turn(90)
        turtle → forward(laenge * 2)
        turtle → right turn(90)
        turtle → forward(laenge)
        turtle → right turn(90)
        turtle → forward(laenge)
        turtle → left turn(90)
        turtle → forward(laenge)
        turtle → left turn(90)
        turtle → forward(laenge * 2)
        turtle → left turn(90)
        turtle → forward(laenge * 3)
        turtle → left turn(90)
        turtle → forward(laenge * 4)
        turtle → left turn(90)
    end for
end function
```


CyL 4.2.2 Mäanderband

Lassen Sie die Kinder zum Ende der Stunde selbst Mäander zeichnen und ausprobieren. Durch die Freiarbeit vertiefen die Schülerinnen und Schüler ihr Wissen und wenden das Gelernte entsprechend ihrem Können an.

Variante: Nicht nur Mäander lassen sich mit Rechenvorgängen darstellen. Auch beliebige n-Ecke können mit schlauen Merkkästen leicht variieren. Hier verwenden wir den Rechenvorgang Division.

Lösungsbeispiel:



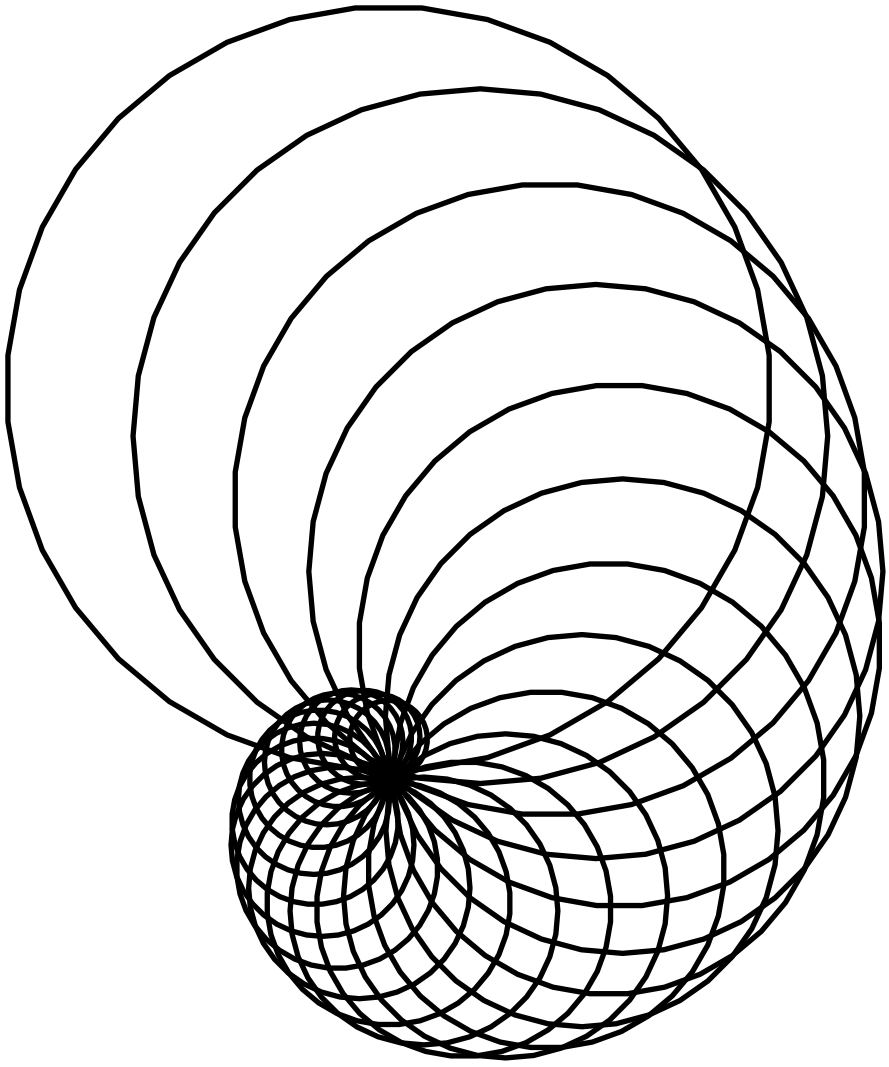
CyL 4.2.2 6-Eck

Hinweis! Gerne können Sie den Kindern ab diesem Zeitpunkt auch zielgerichtet Knobelaufgaben geben, wie z.B. eine quadratische Schnecke oder das Haus vom Nikolaus.

2.3 Ausblick

Geben Sie den Kindern einen kleinen Ausblick auf die kommende Stunde und zeichnen Sie eine Spirale an die Tafel. Dies werden sie in der nächsten Unterrichtseinheit lernen.

4



Gesamtwortschatz der Turtle (Stand Tutorial 4)	
turtle → pen up	Die Turtle nimmt den Stift hoch.
turtle → pen down	Die Turtle setzt den Stift ab.
turtle → left turn (90)	Die Turtle dreht sich um 90° nach links.
turtle → right turn (180)	Die Turtle dreht sich um 180° nach rechts.
turtle → forward (200)	Die Turtle läuft um 200px nach vorne.
turtle → back (100)	Die Turtle läuft um 100px rückwärts.
turtle → set pen color (colors -> red)	Die Turtle wechselt die Farbe auf „rot“.
turtle → fast	Die Turtle zeigt gleich das fertige Bild.
turtle → set speed (400)	Die Turtle läuft in vorgegebener Geschwindigkeit.
turtle → set pen color (colors->random)	Die Turtle wechselt die Farbe nach dem Zufallsprinzip.
for 0 ≤ i < 9 do	Mit der For-Schleife, auch Bruno-Schleife genannt, können Befehle beliebig oft wiederholt werden.
var laenge := 50	Im Nele-Merkkasten können Variablen als Platzhalter eingesetzt werden.

Von Blüten und Spiralen

Mit der fünften Unterrichtseinheit gehen die Schülerinnen und Schüler tiefer in die Systematik des Programmierens ein. Hauptaugenmerk und Lernziel ist es, den Begriff des Zuweisungsoperators kennenzulernen und die Bedeutung dahinter zu verstehen.

Für die Aufgabenstellung im Curriculum „Von Blüten und Spiralen" ist es notwendig, dass sich die von den Kindern eingesetzten Merkkästen dynamisch verändern.

Dazu werden die Kinder wieder sowohl am Bildschirm programmieren, als auch offline die Aufgabenstellung ausprobieren und „nachfühlen".

Heraus kommen am Ende vielfältige Kreationen von Spiralen und Blüten, die den ahnungslosen Betrachter in Staunen versetzen.

Überblick

Zeitaufwand	90 Minuten
Technik	touchfähige Geräte, WLAN (auch am PC mit LAN)
Methoden	Gruppenarbeit, Frage und Antwort, Simulation
Vorkenntnisse	Curriculum 1 bis 4

Kompetenzen

Die Schülerinnen und Schüler ...

- ¬ festigen ihr Wissen und ihre Umsetzungskompetenz aus den letzten Stunden, indem sie sowohl die Bruno-Schleife, als auch den Nele-Merkkasten immer wieder einsetzen und neue Rechenvorgänge verwenden.
- ¬ lernen, wie sie Schleifen in Schleifen einbinden.
- ¬ lernen, dass Variablen in einem Skript neue Werte zugewiesen werden können.
- ¬ wissen am Ende, wie sie es schaffen, eine Spirale zu programmieren.

Ablauf

Phase	Aufgabe	Methode	Zeit
Reflexion	Wiederholung der letzten Stunde	F & A	10‘
Arbeitsphase	Vom Quadrat zur Spirale	Gruppenarbeit & Plenum	70‘
	Blütenkreationen		15‘
	Die aufwärtsdrehende Spirale		30‘
Ausblick	Der Blumengarten	Hausaufgabe	5‘

Unterrichtsverlauf

Vorbemerkung

Der nun folgende Abschnitt verwendet zwar nur einen einfachen Rechenvorgang, dafür ist aber die Struktur komplexer und die Kinder lernen die Bedeutung des Zeichens := kennen. Der sogenannte Zuweisungsoperator ist eine Anweisung im Programm, durch die einer Variable ein neuer Wert zugewiesen wird. Ist eine Variable im Skript bereits klar benannt, kann sie in der Folge einen anderen Wert annehmen. Dies hört sich komplex und kompliziert an, ist aber in der Umsetzung und Anschauung für Kinder gut zu verstehen.

Phase 1 | Reflexion

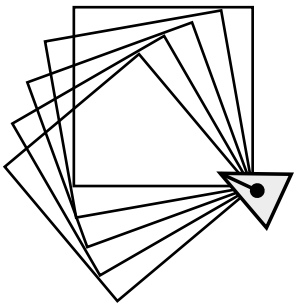
Starten Sie wie gewohnt damit, noch einmal die letzte Unterrichtseinheit Revue passieren zu lassen. Tragen Sie gemeinsam mit den Kindern die Aufgaben und Lösungen zu den komplizierten Mäandern zusammen. Reflektieren Sie noch einmal, warum die Turtle schlau ist und was sie deshalb machen kann. Zeichnen Sie gegebenenfalls den Mäander an die Tafel und gehen Sie die jeweiligen Rechenvorgänge mit den Kindern durch.

Leiten Sie dann in die nun anstehende Arbeitsphase über. Kündigen Sie den Kindern an, dass Sie heute mit einem ganz einfachen Rechenvorgang weitermachen werden: Dem Addieren. Ergebnis werden wunderbare Blütenmuster sein.

Phase 2 | Arbeitsphase 1

2.1 Vom Quadrat zur Spirale

Beginnen Sie langsam und starten Sie mit einer recht einfachen Aufgabe. Zeichnen Sie folgendes Tafelbild an und stellen Sie die untenstehende Aufgabe.



!! Aufgabe:

Zeichnet diese kleine Spirale aus Quadraten nach. Verwendet dazu die Bruno-Schleife und einen Nele-Merkkasten für die Seitenlänge. Die Seitenlänge soll 100 Pixel betragen.

Geben Sie den Kindern Zeit, sich selbst eine Lösung zu überlegen und geben Sie nach und nach individuelle Hilfestellungen. Mit dem erlangten Wissen aus den letzten Unterrichtsstunden müsste diese Aufgabe recht gut gelöst werden können. Dies ist ein guter Moment, um zu sehen, ob alle Schülerinnen und Schüler die vorangegangenen Lernziele erreicht haben. Sammeln Sie nach einer gewissen Zeit die Lösungswege gemeinsam im Plenum mit der ganzen Klasse.

Folgende Fragestellungen helfen Ihnen dabei. Dokumentieren Sie die Antworten nach Bedarf an der Tafel.

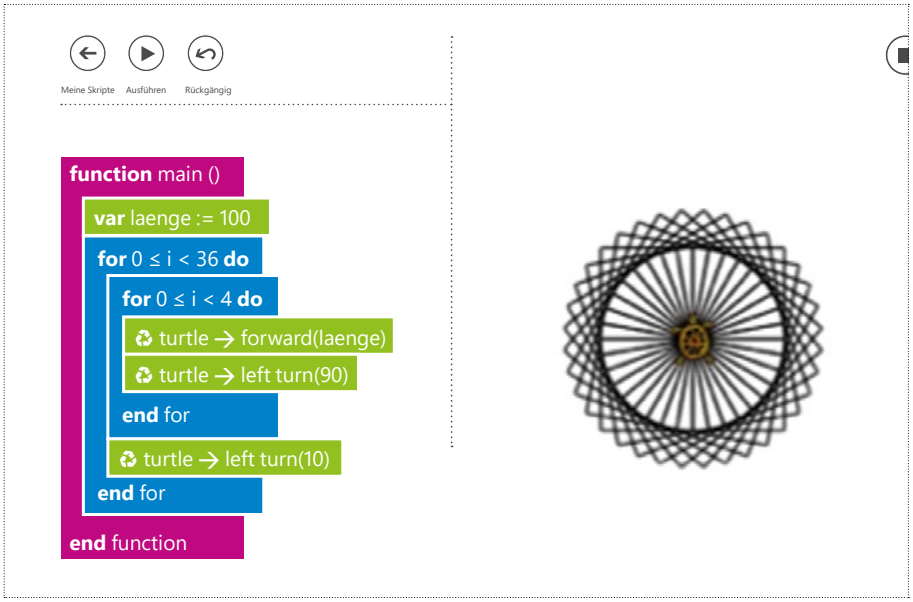
- Woraus besteht die Spirale? (Antwort: aus Quadraten)
- Wie lauten die Codezeilen für eine Spirale?
  (Antwort: Bruno-Schleife mit Forward und Turn Befehlen)
- Und wie machen wir aus einem Quadrat, mehrere Quadrate?
  (Antwort: Noch eine Bruno-Schleife)
- Wo muss die Bruno-Schleife angesetzt werden? (Antwort: außen um die Quadrat-Schleife)

Machen Sie hier ruhig eine kurze Pause und lassen Sie alle Schülerinnen und Schüler das Zusammengetragene ausprobieren. Sie werden schnell merken, dass noch etwas fehlt, um die Quadrate zu „verrücken".

- Was müssen wir nun verändern, damit die Quadrate nicht übereinander liegen, sondern leicht versetzt sind? (Antwort: Turtle muss sich drehen.)

¬ Wann muss die Turtle sich drehen? (Antwort: Nach jedem fertigen Quadrat.)
¬ Wo muss der Befehl dann stehen:
(Antwort: nach der Quadratschleife, aber innerhalb der zweiten Schleife)

Lösungsbeispiel



CyL 5.2.1 Spirale

Lassen Sie die Kinder im Anschluss überlegen, welche Zahl in der äußeren Spirale stehen muss, damit die Spirale einmal ganz herum geht. Hier können die Kinder erst überlegen, oder lieber ausprobieren.

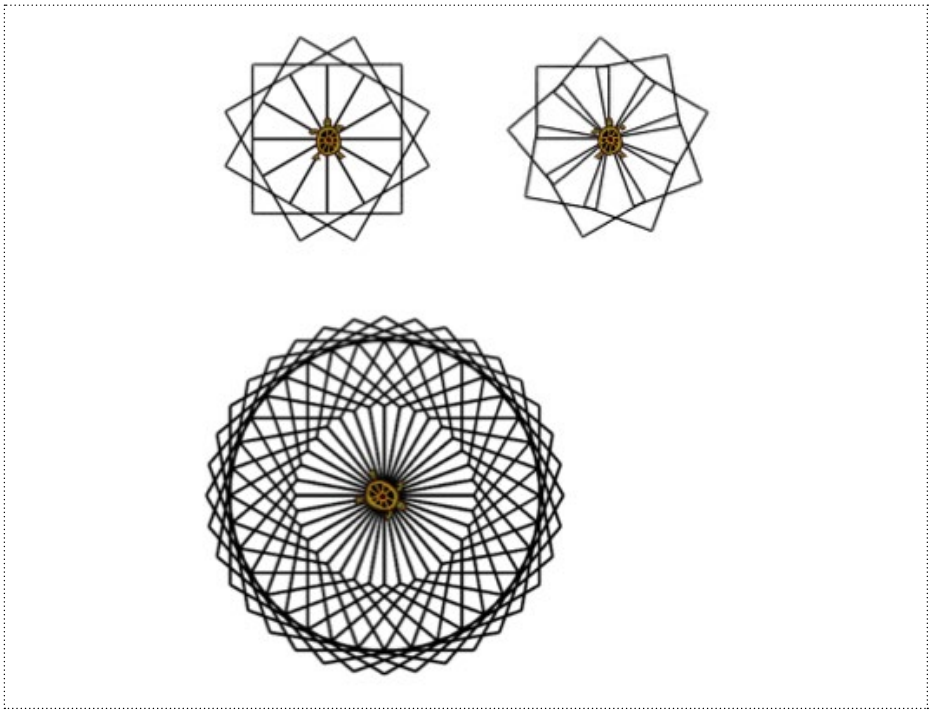
Die Lösung in unserem Beispiel ist: 36-mal, denn so ergibt die Drehung von 10 Grad nach der inneren Bruno-Schleife einen Kreis. Die Turtle ist einmal herumgelaufen.

2.2. Blütenkreationen

Lassen Sie die Kinder nun ein wenig in den Gradzahlen variieren, so dass wunderschöne Blüten aus Spiralen entstehen. Regen Sie die Kinder an, auch einmal andere Grundformen auszuprobieren (z.B. ein Sechseck). Nehmen Sie sich für diesen Abschnitt gerne eine Weile Zeit, damit die Schülerinnen und Schüler von einem Wow-Effekt zum anderen kommen.

Variante: Wenn einige Schülerinnen und Schüler recht fix sind, die Blüten zu programmieren, können Sie ihnen die Aufgabe geben, möglichst viele unterschiedliche Blüten zu zeichnen. Erinnen sie auch nochmal an den Befehl „SET PEN COLOR“, um bunte Blüten zu malen.

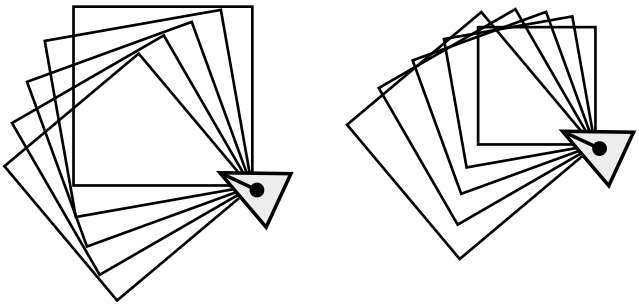
Lösungsbeispiele



2.3 Die aufwärtsdrehende Spirale

Nach dieser kreativen Phase sammeln Sie die Aufmerksamkeit der Kinder wieder, denn jetzt fehlt nur noch ein kleiner aber bedeutender Schritt und die Blüten von eben, werden zu Spiralen.

Zeichnen Sie folgende aufwärtsdrehende Spirale neben die eingangs programmierte Spirale an die Tafel.



Fragen Sie die Kinder:

- Was ist an dieser neuen Spirale anders als an der eingangs programmierten Spirale? (Antwort: die Quadrate werden immer größer.)

Die Kinder sollen nun erst einmal ausprobieren, ob Sie von selbst auf die Lösung kommen. Lassen Sie ihnen hierfür ruhig 10 Minuten Zeit. Danach besprechen Sie gemeinsam das Problem und mögliche Lösungen. Wahrscheinlich werden die Schülerinnen und Schüler einige Lösungswege ausprobiert haben. Eine richtige Lösung ist selten. Sammeln Sie daher die unterschiedlichen Lösungsversuche, reflektieren Sie mit allen, was an dem jeweiligen Weg nicht funktioniert hat.

Erarbeiten Sie nun gemeinsam die richtige Lösung. Stellen Sie dazu wieder eine Reihe Fragen:

- Was verändert sich von Quadrat zu Quadrat? (Antwort: Die Seitenlänge)
- Wieviel verändert sich die Seitenlänge? (Antwort: Zum Beispiel jeweils um 10 Pixel)
- Ändert sie sich jedes Mal um gleich viel Pixel? (Antwort: Ja)

Schieben Sie nun eine kleine Offline-Übung ein.

Holen Sie das Kind aus der Unterrichtseinheit „Der Nele-Merkkasten“ nach vorne, in unserem Beispiel Nele und bitten Sie noch drei weitere dazu, z.B. wieder Paul als Turtle und Thu Thao als Rechenexpertin sowie ein neues Kind, sagen wir Felix. Felix bekommt lauter Klebezettel in die Hand, auf denen jeweils eine der Zahlen 110, 120, 130, 140, 150 steht.

Nun stehen schon 4 Kinder vorne, was zeigt, dass wir mittlerweile schon sehr komplexe Dinge programmieren und die Kinder bereits einige Prinzipien gelernt haben.

Bruno als Schleife lassen wir in diesem Beispiel sogar weg.
Zuerst gibt Nele als Merkkasten die Seitenlänge 100 an Paul, die Turtle weiter.

Paul soll nun als erstes das Quadrat mit einer Seitenlänge von 100 Millimeter an die Tafel malen. (Hinweis: Hier darf etwas gemogelt werden und wir geben vor, dass Paul als Turtle ein Quadrat malen kann ohne die einzelnen Schritte mit der Bruno-Schleife angesagt zu bekommen. Auch entspricht die Seitenlänge nicht wirklich 100 Millimeter, sondern wird vielmehr so groß gezeichnet, dass es alle sehen können.)

Nun wollen wir, dass Paul ein um 10 Millimeter größeres Quadrat daneben malt. Da kommen Thu Thao und Felix ins Spiel. Thu Thao schiebt sich zwischen Nele und Paul und rechnet für den nächsten Durchgang 10 Millimeter zur Länge hinzu, das Ergebnis (110 Millimeter) teilt sie nun Paul mit. Paul zeichnet ein Quadrat das 110 Millimeter Seitenlänge hat. Hat er das Quadrat gezeichnet, läuft Felix zu Nele und klebt ihr die Zahl 110 an. Und der Durchlauf beginnt von vorne. Nele flüstert 110, Thu Thao addiert 10 und gibt das Ergebnis (120) an Paul weiter. Dieser zeichnet. Felix zückt den zettel 120 und klebt ihn Nele an. Und so weiter.

Was brauchen wir also alles zusätzlich zum eingangs programmierten Skript für die Spirale?

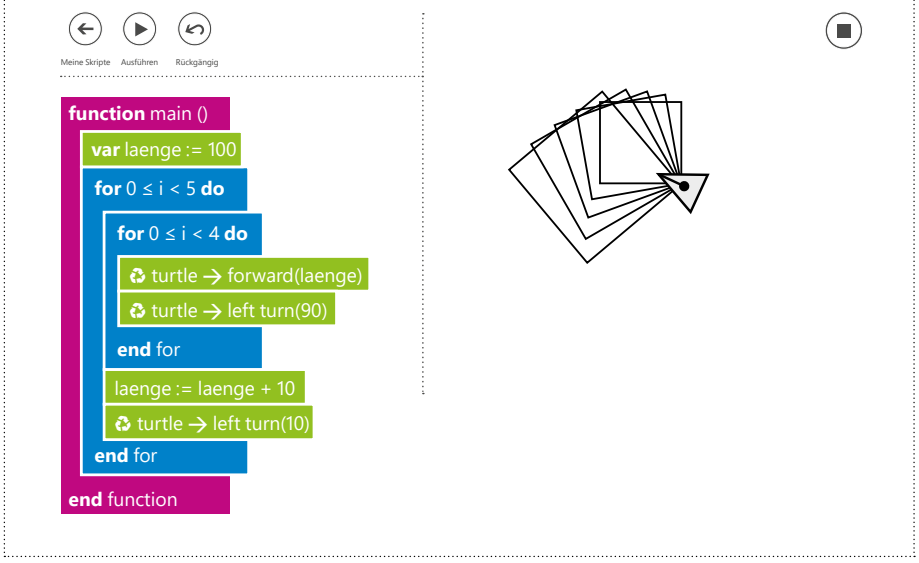
¬ Einen Nele-Merkkasten, also eine Variable für die Seitenlänge.
¬ Einen Thu Thao Rechenvorgang der Addition.
¬ Einen Felix-Klebezettel, der aus der alten Länge eine neue macht.

Nach dieser Übung lenken Sie den Blick der Kinder auf das Zeichen **:=.** Erläutern Sie, dass es sich hier nicht um ein Gleich-Zeichen wie aus der Mathematik handelt, sondern dass dies ein Zuweisungsoperator ist. Genauso wie im Offline Beispiel bleibt Nele im Skript Nele (also die Laenge), verändert sich aber durch Thu Thao, den Rechenvorgang, und bekommt durch den Felix-Klebezettel einen neuen Wert zugewiesen.

Lassen Sie die Kinder dies nun in ihr Skript übertragen. Begleiten Sie die Kinder dabei und gehen Sie die einzelnen Schritte nochmal durch.

5

Lösungsbeispiel



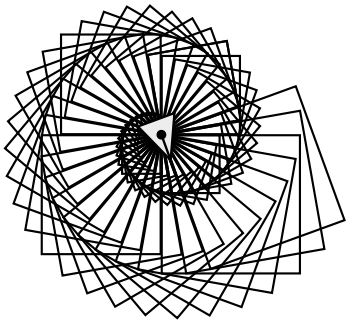
CyL 5.2.3 Spirale 1

Haben die Schülerinnen und Schüler dies erfolgreich geschafft, können sie ausprobieren, was passiert, wenn man Werte ändert, Gradzahlen erhöht, andere Rechenvorgänge einbaut und so weiter. Lassen Sie die Kinder zum Ende nun ein wenig experimentieren.

Hinweis! Benötigen die Kinder noch eine Übungsphase oder wollen sie weitere Aufgaben, können sie hier noch zusätzliche Aufgaben einfügen. Einen Kreis in Spiralform, ein Sechseck als Spirale. Alternativ können Sie diese Übungsphase auch an den Anfang der nächsten Stunde setzen.

Meine Skripte Ausführen Rückgängig

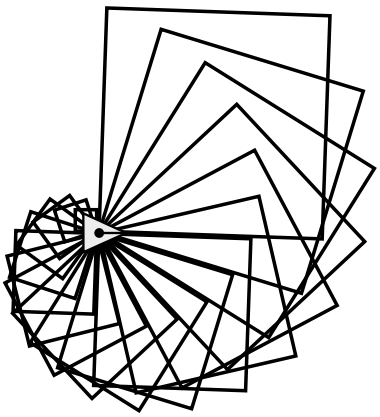
```
function main ()
    var laenge := 20
    for 0 ≤ i < 60 do
        for 0 ≤ i < 4 do
            turtle → forward(laenge)
            turtle → left turn(90)
        end for
        turtle → laenge + 2
        turtle → left turn(10)
    end for
end function
```



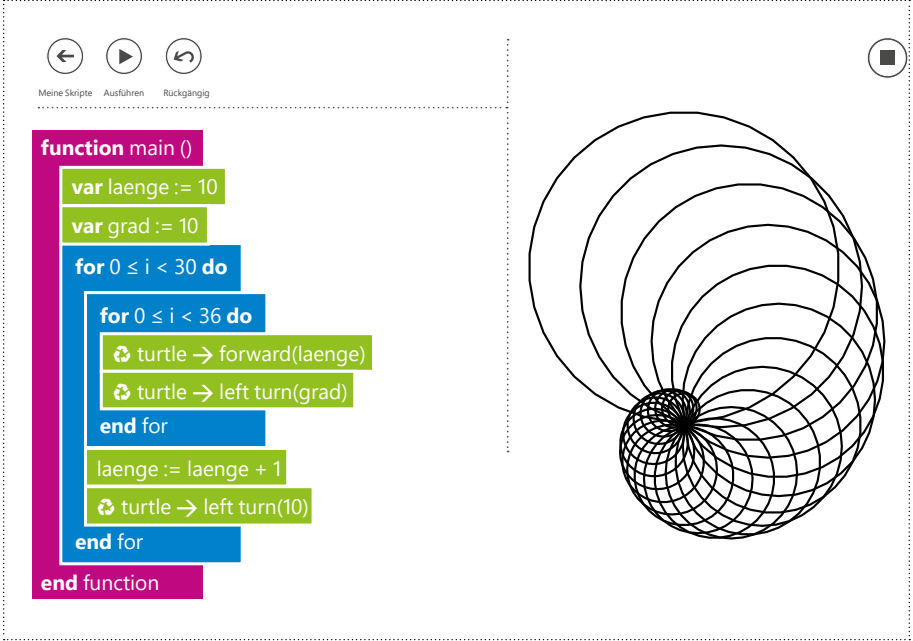
CyL 5.2.3 Spirale 2

Meine Skripte Ausführen Rückgängig

```
function main ()
    var laenge := 100
    for 0 ≤ i < 16 do
        for 0 ≤ i < 4 do
            turtle → forward(laenge)
            turtle → left turn(90)
        end for
        laenge := laenge + 10
        turtle → left turn(18)
    end for
end function
```



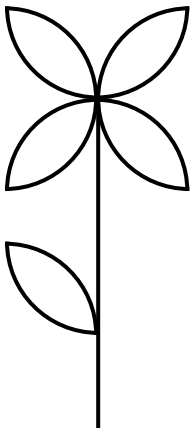
CyL 5.2.3 Spirale 3



CyL 5.2.3 Spirale 4

2.4 Ausblick

Geben Sie den Kindern einen kleinen Ausblick auf die kommende Stunde. Zeichnen Sie folgende Blume an die Tafel. Wer mag, kann sich zuhause schon einmal überlegen, wie man diese Blume wohl programmieren könnte.



Einblicke in unsere Praxis

Es ist großartig zu sehen, mit welcher Neugierde und unaufhörlichem Forscherdrang die Kinder an das Programmieren herangehen. Ganz ohne Berührungsängste.

Jutta Schneider – Projektleitung Code your Life

Die ersten Codezeilen werden geschrieben

Gespannte und erwartungsvolle Gesichter beim Auftakt der Initiative

Der Auftakt-Event

Wir schreiben Code

„Wow, deine Schildkröte zeichnet ja sogar eine Spirale!". Mia ist 12 Jahre alt und kommt aus einer Berliner Grundschule. Sie sitzt mit zwei Klassenkameraden vor einem Tablet und schaut gespannt zu, wie sich langsam auf dem Bildschirm die Quadrate zu einer Spirale zusammenfügen. Sie hat gerade einfache Befehle einer Programmiersprache gelernt und ist völlig fasziniert von den Ergebnissen.

Nebenan rollt der rot leuchtende Sphero-Ball fast von der Bühne. „Stopp. Du musst weniger Geschwindigkeit eingeben, sonst geht er uns noch kaputt", ruft Paul seinem Team zu. Ganz praktisch erlebt er so, wie sich der geschriebene Code sofort in der Realität auswirkt. „Man muss schon ganz genau schreiben, wie der Sphero rollen soll. Wenn ich einen Tippfehler mache, bewegt der sich null", erklärt Paul. Trotz der Herausforderung schaffen die Jungs es am Ende doch, den Sphero im Kreis rollen zu lassen und quittieren sich gegenseitig ihren Erfolg mit einem High-Five.

Ein Tag voller Inspiration

Es war viel los an diesem Tag, Anfang Dezember in den Räumen von Microsoft Berlin. Rund 70 Kinder und ihre Lehrkräfte waren gekommen, um gemeinsam mit Microsoft und dem 21st Century Competence Center den Auftakt zur Initiative Code your Life zu starten.

In verschiedenen Workshops hatten die Kinder die Möglichkeit, Programmiersprachen kennenzulernen, sich spielerisch in Computerprogramme hinein zu versetzen und einen Einblick in die faszinierende Welt der IT zu bekommen. Von eingeladenen Experten und Mentoren haben die Kinder erfahren, wie man eine App oder ein Computerspiel macht und sahen dabei, dass die sogenannten Nerds echt coole Leute sind.

„Das war cool. Können wir das auch in der Schule machen?", war ein eindeutiges Feedback der Sechstklässler. Die Antwort lautete: Ja. Im darauffolgenden Schulhalbjahr kam das Team von Code your Life regelmäßig in die Klassen und gab den Kindern so die Möglichkeit, tiefer in das Thema einzusteigen. Es blieb also weiterhin spannend für Paul, Mia und ihre Freunde.

Die Turtle läuft

Die erste Stunde

Es ist Donnerstag, 9:50 Uhr und zwanzig Augenpaare sind auf uns gerichtet. Gespannt und mit großer Vorfreude haben sich die Schülerinnen und Schüler unserer ersten Pilotklasse zusammengefunden. Eigentlich wäre jetzt Hofpause, aber die Kinder wollen lieber programmieren. Auf dem Unterrichtsplan stehen heute TouchDevelop und Programmierung in Logo.

„Stellt euch vor, ich bin eine Schildkröte und das ist meine Nase.", Thomas, der Dozent, stellt sich vor die Klasse und spielt die Turtle. Er fordert die Kinder auf, ihm Befehle zu geben, so dass er sich als Schildkröte vorwärts oder rückwärts bewegt. Da läuft er schon einmal gegen den Tisch oder die Tafel, wenn nicht rechtzeitig Stopp gerufen wird oder die Schrittzahl nicht stimmt.

Das bunte Quadrat

Langsam bekommen die Kinder ein Gefühl dafür, welche Befehle und welche Worte wichtig sind, um die Thomas-Turtle in live und später auch die animierte Turtle auf den Surface-Geräten zum Laufen und Malen zu bringen. Ganz spielerisch wird so der grundlegende Wortschatz der Programmiersprache Logo für die Kinder erfahrbar.

Und tatsächlich: Ganz ohne Berührungsängste mit der Anwendung auf den Geräten schaffen die Kinder es, dass die Turtle auf ihrem Bildschirm ein Quadrat zeichnet. Sogar mit verschiedenfarbigen Seitenlinien. Kinder, Dozenten und auch die Klassenlehrerin sind begeistert! Das macht Lust auf mehr.

Die Turtle in Aktion

Die Bruno-Schleife

Zeile um Zeile

Unsere zweite Lerngruppe setzt sich aus Schülerinnen und Schülern unterschiedlicher Jahrgangsstufen zusammen. Von der 4. bis zur 6. Klasse lernen die Kinder mit uns am Nachmittag das Programmieren. Auch hier wird die Turtle zum Malen und Zeichnen gebracht.

Heute sind wir für die zweite Unterrichtseinheit gekommen. Das Quadrat kennen die Kinder also schon. Aber schaffen Sie es auch, dass die Turtle ein Sechseck, oder sogar Sechsunddreißig-Eck malt? Gemeinsam denken wir nach, wie groß die Parameter sein müssen. Alle legen fleißig los und tippen Zeile um Zeile Code. Bei der 18. Codezeile fängt Bruno an zu meckern. Das wäre zu anstrengend, ob man nicht etwas wie „Copy und Paste“ oder „Wiederholen“ machen könnte.

Ein neues Programmierprinzip

Die Dozenten grinsen. Wunderbar, wir sind dem heutigen Lernziel ganz nahe. Bruno darf nach vorne kommen und gemeinsam mit Tara das Computerprogramm spielen. Sie stellen sich hinter den Dozenten Thomas, der mal wieder die Schildkröte spielt.

Die Gruppe überlegt, wie sie der Turtle mit so wenig Zeilen wie möglich sagen kann, dass sie ein Sechseck gehen soll. Bruno fängt an, Thomas Befehle zu geben: Forward (100), Left Turn (60).

Tara klopft Bruno immer wieder auf die Schulter. Insgesamt sechs Mal. Und siehe da, Thomas, die Turtle ist ein Sechseck gelaufen. Toll, die Bruno-Schleife ist geboren und wir haben wieder ein Programmierprinzip gelernt. Mit der „For“-Schleife, ab sofort hier Bruno-Schleife genannt, können ganz leicht Codeabfolgen beliebig oft wiederholt werden.

Kreativität und Problemlösung auf höchtem Niveau.

Mirobot meets Turtle

Ein YouthSpark Live Event

Am 28. Mai 2015 gab es einen ganz besonderen Event. Code your Life hatte drei Berliner Schulklassen eingeladen, einen ganzen Projekttag lang in den Räumen von Microsoft Berlin zu programmieren. Diesmal wollten wir zeigen, wie viel schöne Kunst mit Computer-Code geschaffen werden kann. Dazu nutzten wir den zeichnenden Miniroboter Mirobot und programmierten eindrucksvolle Muster.

Die größte Herausforderung des Tages war es, erst einmal die kleinen Miniroboter gemeinsam mit den Schülerinnen und Schülern zusammenzubauen. Wo muss der Motor sitzen? Wozu brauchen wir eine Antenne und wie verbinden wir die Kabel? Schritt für Schritt arbeiteten sich die Jugendlichen durch die Bauanleitung des Mirobots. Dabei mussten sie sowohl Fingerspitzengefühl als auch Geduld und logisches Denken beweisen.

Ein Miniroboter in Aktion

Parallel dazu konzentrierten sich die Kreativteams darauf, mit viel Entdeckergeist, Nachdenken und Experimentieren, dem Programmieren näher zu kommen. Aus Halbkreisen, Mäandern und Spiralen wurden eigene Kunstwerke programmiert. An eine Pause wollte keiner der Schülerinnen und Schüler denken. Mit unglaublichem Enthusiasmus wurde Zeile um Zeile Code geschrieben.

Schließlich hat sich all das Engagement gelohnt. Am Ende hielten die Teams fünf fertige Roboter in den Händen und konnten die unterschiedlichsten Muster programmieren. Da gebührt den Schülerinnen und Schülern ein großer Dank für die fleißigen Hände und kreativen Ideen! Die Roboter nehmen nämlich seither im Rahmen von Code your Life einen wichtigen Platz ein. Und wer weiß, vielleicht haben wir mit diesem Event den Grundstein dafür gelegt, dass es zukünftig mehr Miniroboter in unseren Schulen gibt. Zum Beispiel im Kunstunterricht.

Zu Gast bei Microsoft

Offline mit Zettel und Stift

Es ist Dienstag früh und die Türen von Microsoft öffnen sich heute für 21 Schülerinnen und Schüler aus Pankow. Mit großen Augen werden die Räume begutachtet und ganz vorsichtig die Surfaces aufgeklappt. Auf der großen Leinwand ist zu lesen, was die Kinder heute erwartet: Code your Life.

Was ist eigentlich Programmieren und was braucht man dazu? Einen Computer? Eigentlich schon. Aber um zu verstehen, welche Sprache ein Computerprogramm versteht, lassen wir erst einmal die Tablets aus und „spielen" selbst Entwickler und Computerprogramm.

Rücken an Rücken sitzen die Kinder in Zweierteams zusammen. Einer spielt den Entwickler und bekommt ein Blatt mit einem Bild bzw. Muster darauf. Der Andere ist das Computerprogramm und soll nun exakt das gleiche Bild zeichnen, aber ohne es zu sehen. Dafür muss der Entwickler ihm ganz konkret beschreiben, was zu zeichnen ist. Er muss sogenannte Befehle geben. Die Kinder merken schnell, dass man nicht einfach sagen kann: „Male ein Auto", denn für jeden sieht ein Auto anders aus.

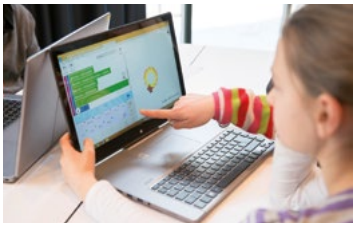
Muster programmieren

Zum Glück hat ein Bildschirm ähnlich wie ein kariertes Papier Hilfspunkte, nämlich die Bildschirm-Pixel, mit denen man sehr genau sagen kann, wie lang oder kurz ein Strich sein soll. So können wir es schaffen, dass das „echte" Computerprogramm für uns Bilder malt. Also: Computer an und los geht's.

Nachdem die Kinder den grundlegenden Programmierwortschatz und das Prinzip der For-Schleife kennen gelernt hatten, war der Weg frei, Bilder und Muster auf den Bildschirm zu programmieren. Die letzte Unterrichtseinheit des Tages nutzten wir, um mit den bisher gelernten Befehlen „zu spielen". Ziel war es, immer wieder neue Muster zu erfinden. Die Kinder hatten großen Spaß und auch die Lehrerin war hin und weg. Als IT-Verantwortliche der Schule will sie auf jeden Fall mit ihren Schülerinnen und Schülern weitermachen.

Wir freuen uns, wieder eine Schule für das Programmieren begeistert zu haben und dass wir so dazu beitragen, dass mehr Kinder schon im Grundschulalter spielerisch an Informatik herangeführt werden.

Stolze Gesichter
im Microsoft Center



Programmieren „offline“ mit Zettel und Stift


CODE your life
Microsoft YouthSpark
CODE your life


für deine Zukunft!

Code your Life beim Girls' Day

Zu Gast bei der Bundeskanzlerin

Im April wurde uns eine ganz besondere Ehre zuteil: Zum Auftakt des Girls' Day 2015 im Kanzleramt mit Bundeskanzlerin Angela Merkel durfte auch Code your Life mit dabei sein. Der ganze Tag war davon geprägt, zu zeigen, wie spannend und vielfältig technische und naturwissenschaftliche Berufe sind. „Da sind riesige Möglichkeiten", ermunterte Bundeskanzlerin Angela Merkel die anwesenden Mädchen in ihrer Begrüßungsrede.

Am Stand von Code your Life konnten die Mädchen dies auch gleich austesten und mit uns gemeinsam programmieren. Dabei sind wir besonders stolz auf die drei Mädchen, die an unserem Stand in kürzester Zeit zu Junior Codern geworden sind, um dann der Bundeskanzlerin zu erklären, wie man Textnachrichten auf das LED-Board programmiert, wie man sie zum Leuchten bringt und über den Bildschirm bewegen lässt.

Ein toller Tag, an dem sich wieder gezeigt hat, dass Programmieren sehr wohl etwas für Mädchen ist. Die schönste Nachricht, die von den Mädchen programmiert wurde und im Kanzleramt über das LED-Board flimmerte, kann auch als Fazit des gesamten Girls' Day Auftakts gesehen werden: „IT is cool" stand in großen roten und blauen Buchstaben auf dem Screen. Wer weiß, vielleicht sehen wir einige der Mädchen bald einmal als Nachwuchsprogrammierer wieder.

Über 70 Mädchen programmieren

Weiter ging es am darauffolgenden Tag mit unserem eigenen Event zum Girls' Day. Im großen Atrium von Microsoft Berlin startete der Tag mit begrüßenden Worten von Christina Hadulla-Kuhlmann, Ministerialrätin im Ministerium für Bildung und Forschung, die für die Mädchen auch Tipps zur Berufsfindung im Gepäck hatte. Anschließend legten sich die Schülerinnen ins Zeug und programmierten mithilfe von Filtern und Abfragen ihre eigenen Twitterwalls zum Thema „Frauen und Technik". Dabei twitterten sie fleißig eigene Tweets über den Tag unter dem Hashtag #CYL.

Nebenbei konnten die Mädchen beim Geek Dating mit erfolgreichen Frauen aus der IT-Welt einen Blick hinter die Kulissen werfen und erfuhren anhand ganz persönlicher Geschichten, welche spannenden Jobs es eigentlich im Technologie-Bereich gibt. Eine Fundgrube an Inspiration für die Schülerinnen.

So zieht Sarah, eines der Mädchen, ein ganz persönliches Resümee: „Ich habe heute nicht nur meine ersten Zeilen Code geschrieben, sondern auch gelernt, dass es verschiedene Wege gibt, um einmal bei Microsoft und Co. zu arbeiten. Mal sehen, ob ich meinen Junior Coder Button vom heutigen Tag einmal zu einem Expert Coder mache. Lust habe ich auf jeden Fall."

Bundeskanzlerin Angela Merkel bei der Eröffnungsrede zum Girls' Day 2015

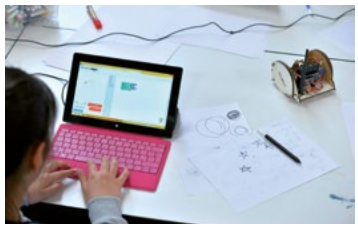
Summer Coding-Camp

Drei Tage Coding, Sport und Spaß

Über 80 Schülerinnen und Schüler aus dem ganzen Bundesgebiet kamen vom 08.-10. Juni 2015 in die PerspektivFabrik bei Brandenburg, um gemeinsam mit uns das erste Summer Coding-Camp zu einem wahren Highlight der Pilotphase von Code your Life zu machen. Dabei wurde die Turnhalle der Location am See kurzerhand mit Wifi und Surface Geräten in ein digitales Klassenzimmer verwandelt.

Sogar der Microsoft Tech Truck fuhr vor und breitete viele Gadgets und Zukunftstechnologien aus.

Aufgeteilt in verschiedene Teams machten sich die Schülerinnen und Schüler sogleich hochmotiviert ans Programmieren und bewiesen dabei nicht nur Kreativität und ihre Fähigkeiten zum Logischen Denken, sondern unterstützen sich ganz wunderbar untereinander, da jede Lerngruppe unterschiedliche Vorkenntnisse mitbrachte. So erlebten die gemischten Teams drei ereignisreiche Tage mit spannenden Challenges an verschiedenen Stationen, aus denen viele unterschiedliche Ergebnisse und Wow-Effekte entstanden sind.



Gemeinsam lernen, spielen und programmieren beim Summer Coding-Camp.

Vom Junior Coder zum Coding-Experten

Animierte Schildkröten malten Irrgärten in TouchDevelop, der Sphero-Ball verwandelte sich in eine Disko-Kugel und fuhr in Kreisen und Spiralen durch die Halle und die selbst programmierten TwitterWalls zeigten interessante Menschen aus der Welt der Technik und Informatik. Auf dem großen LED-Board waren die Teamnamen zu lesen, der DIY-Roboter Mirobot zauberte Kunst auf große Papierbögen, es wurde mit AntMe! das stärkste Ameisenvolk gekürt und die Schüler bauten sogar das perfekte Camp der Zukunft in Minecraft nach!

Beeindruckend waren das unermüdliche Engagement und die hohe Konzentration der Schülerinnen und Schüler. Aus den kleinen Junior Codern wurden zum Ende hin richtig große Coding-Experten, die vielleicht in Zukunft als Trainer ihren Peers das Programmieren beibringen.

Coding-Konzepte

Haben Sie Lust auch andere Tools oder Anwendung mit ihrer Lerngruppe auszuprobieren? Im Folgenden stellen wir interessante Ideen vor, die den Code your Life-Teilnehmerinnen und Teilnehmern aus der Pilotphase viel Freude bereitet haben und eine optimale Ergänzung zu den Curricula mit Logo-Grafiken darstellen.

Nutzen Sie zum Beispiel den Mirobot, um die Muster der Kinder auch auf Papier zu bringen, lassen sie die Kinder eine eigene Twitterwall programmieren oder bringen Sie mit dem Sphero Ball einen leuchtenden Roboterball ins Spiel. Lassen Sie sich inspirieren und entdecken Sie mit uns die Vielfalt der Möglichkeiten, wie Kinder Programmieren lernen können.

TwitterWall

Überblick

Thema	Social Media & Datenanalyse einmal anders
Altersgruppe	14-18 Jahre
Einsatzgebiet	Nahezu jedes Thema und jedes Fach ist denkbar. Insbesondere politische und gesellschaftliche Themen eignen sich zur Analyse.
Vorkenntnisse	Nicht zwingend notwendig
Kurzüberblick	Schülerinnen und Schüler programmieren ihre eigenen TwitterWalls. Sie nutzen Twitter als Quelle für unterschiedlichste Informationen zu einem gewählten Thema, die sie mithilfe selbst zusammengestellter Filter analysieren. Dazu erstellen sie eigene Datenbanken, hantieren mit logischen Operatoren, bauen Verknüpfungen und Abfragen. Mit der TwitterWall können Tweets der letzten sechs Tage gesammelt und je nach gewähltem Filter auf der Wall dargestellt werden.
Technologie	TwitterWall-Coder. Eine von den Entwicklern des 21CCC eigens für die Initiative Code your Life entwickeltes Framework, das in seiner Syntax auf XML basiert und mit einer ansprechenden Darstellungsform und Benutzeroberfläche eine schnelle Einarbeitung ermöglicht.
Szenario	Die Schülerinnen und Schüler erstellen eigene Tweets zu einem bestimmten Thema oder wählen die TwitterWall als kreative Darstellung einer Linksammlung.

Die TwitterWall ist ein ideales Instrument, um sowohl Programmierprinzipien zu vermitteln, als auch Datenanalyse zu thematisieren sowie die Arbeitsprozesse in einer Medienagentur kennenzulernen. Je nach Ressourcen und Lernszenario simulieren die Kinder die Arbeit einer Medienagentur von der Ideengewinnung über die Recherche bis hin zur tatsächlichen Programmierung von Design und Content des Ergebnisses. Dabei können die Schülerinnen und Schüler in die Rolle von Redakteuren, Entwicklern, Designern und Content-Managern schlüpfen.

Um am Ende eine TwitterWall zu haben, die genau das zeigt, was die Lerngruppe sich vorab ausgedacht haben, müssen die Kinder mithilfe von grundlegenden Programmierprinzipien den passenden Inhalt zusammenstellen.

Mit logischen Operatoren wie AND oder OR stellen sie Verknüpfungen her, um ihre Datenbank zu pflegen. Das heißt, sie müssen kombinieren und ausprobieren mit welchen Hashtags sie die besten Ergebnisse erzielen.

Um beispielsweise zu analysieren, wie sich Microsoft aktuell weltweit zum Thema Frauen in der IT engagiert, könnte eine Codezeile wie folgt aussehen: (#Microsoft + #women + #coding) OR (#MSFT + #Frauen + #Programmieren) bzw. (#MSFT) AND (#ComputerGirls)

Mit Abfragen (IF und ELSE) können bestimmte Bedingungen aufgestellt werden, mit denen Ereignisse und Verzweigungen im Programm deklariert werden. So kann zum Beispiel festgelegt werden, dass alle Tweets mit dem Hashtag #Microsoft auf der Twitter-Wall mit großen blauen Sprechblasen erscheinen.

Sie haben Interesse und möchten mehr erfahren?
Kontaktieren Sie uns!

21CCC@fjs-ev.de
030 2938 1680
www.code-your-life.org

Mirobot

Überblick

Thema	Roboter & Logo-Programmierung
Altersgruppe	ab 8 Jahre
Einsatzgebiet	Mathematik, Kunst
Vorkenntnisse	Nicht zwingend notwendig, mathematische Grundlagen zur Winkelberechnung hilfreich
Kurzüberblick	Mithilfe des Roboters Mirobot programmieren bereits Kinder ab 8 Jahren geometrische Figuren und lassen diese von ihm auf Papier bringen.
Technologie	Der Mirobot ist ein DIY-Roboter, der als Bausatz erhältlich ist und mit Kindern gemeinsam zusammengebaut werden kann. Die Programmierumgebung ist webbasiert und kann über den Browser angesteuert werden. Die verwendete Sprache basiert auf Logo und Turtle-Grafiken.
Ressource	www.mirobot.io

Der Mirobot ist die perfekte Ergänzung zur Programmierung von Logo-Grafiken mit TouchDevelop. Dieser kleine Roboter bringt die Grafiken und geometrischen Figuren der Kinder vom Bildschirm auf das Papier. Durch die einfache Handhabung können Kinder schnell und intuitiv einzigartige Muster zeichnen und so anwendungsbezogen ihre praktische Programmiererfahrungen erweitern.

Auf der webbasierten Programmierumgebung setzen die Kinder ihr Skript im Block-Style zusammen und können Parameter und Operatoren anpassen. Die Codezeilen können zuerst am Bildschirm ausprobiert werden, bevor der Mirobot zum Einsatz kommt. So können die Kinder überprüfen, ob das erwartete Ergebnis ihres Codes auch eintrifft. Anschließend verbinden Sie den Mirobot mit ihrem Rechner und lassen ihn das Skript zeichnen. Dabei kommuniziert der Roboter über das WLAN.

Kinder lieben den Mirobot und bringen meist die kunstvollsten Bilder zustande. Der Transfer der gelernten Prinzipien aus der Programmierumgebung TouchDevelop zur Block-Programmierung beim Mirobot wird dabei ohne Probleme geschafft. Schnell erkennen die Kinder, was zu tun ist und legen los. Voller Stolz können sie dann die Ergebnisse ihrer Kreativität, ihres abstraktem und logischen Denkens mit nach Hause nehmen.

Mirobot

Der Mirobot wurde als Kick-Starter-Kampagne von Ben Pirt in Großbritannien entwickelt. Als Vater zweier Kinder war und ist er stets von den Möglichkeiten begeistert, Kindern auf spielerische Weise Technologien nahezubringen. Inspiriert von einem der ersten Logo-Roboter aus den 1970ern und seinen Erinnerung an diesen wertvollen Schatz in manchen Schulen dieser Zeit, entstand sein Traum, ein kostengünstiges und 100%iges Open-Source-Tool zu entwickeln. Mit dem Mirobot hat er diesen Traum in die Tat umgesetzt und den bewährten Logo-Roboter für Schulen und Bildungseinrichtungen wieder salonfähig gemacht.

Sie möchten einen Mirobot ausprobieren?
Schauen Sie einfach auf www.mirobot.io vorbei.

Sie suchen Ideen für den Einsatz des Mirobot?
Kontaktieren Sie uns!

21CCC@fjs-ev.de
030 2938 1680
www.code-your-life.org

Minecraft

Überblick

Thema	Gamicfication
Altersgruppe	ab 8 Jahre
Einsatzgebiet	Mathematik, Geschichte, Biologie, Beteiligungsprojekte, Zukunftsprojekte, Architektur
Vorkenntnisse	Nicht zwingend notwendig
Kurzüberblick	Mit Minecraft erstellen Kinder und Jugendliche eigene Welten, bauen sich ihre Zukunftsvisionen und lösen Aufgaben. Über die Programmierung von Turtle-Robotern als Sidekicks im Spiel können sie dabei schneller und effizienter zum Ziel gelangen.
Technologie	Minecraft ist ein Open-World-Spiel, erhältlich für PC, MAC, Konsolen und als Pocket-Version auch auf dem Smartphone nutzbar. Mit MinecraftEdu wurde bereits eine spezielle Version von Minecraft für Schulen entwickelt, die zusätzliche Features bereithält und an die Bedürfnisse von Bildungssituationen angepasst ist. Nach der Übernahme durch Microsoft Anfang 2016, ist mit einer tatkräftigen Weiterentwicklung der Edu-Version zu rechnen.
Ressource	www.minecraft.net www.minecraftedu.org www.computercraftedu.com

Ohne Zweifel ist Minecraft das derzeit beliebteste Computerspiel. Mädchen und Jungen auf der ganzen Welt erkunden tagtäglich die Spielwelt, bauen Rohstoffe ab und verarbeiten sie weiter, bauen Gebäude und erschaffen eigen Welten, erleben Abenteuer und erfüllen Aufgaben. Kinder und Jugendliche gehen in Minecraft intuitiv lösungsorientiert vor, um die Aufgaben und Gefahren im Spiel zu bewältigen. Lerneffekte stellen sich sehr schnell ein, die Spieler interagieren im Spiel miteinander und erkennen recht schnell wie sie effektiv im Team arbeiten um bessere Ergebnisse zu erzielen.

Minecraft gewinnt immer mehr an Bedeutung in Bildungseinrichtungen, da Pädagogen und Pädagoginnen das Potential der Motivation durch Gamification für ihren Unterricht oder ihr Bildungsziele erkannt haben. Von der politischen Bildung über Zukunftsworkshop und Partizipationsprojekte bis hin zu Minecraft in Biologie oder Geschichte: Die Vielfalt der Einsatzgebiete ist groß und es werden fast täglich neue Best-Practices im Netz vorgestellt.

Dabei muss der Pädagoge selbst kein Minecraft-Routinier sein. Die Handhabung im Spiel beherrschen die meisten Kinder und Jugendlichen selbst und helfen sich untereinander.

Um große und komplexe Gebäude, Gärten oder Welten zu erschaffen, kann in Minecraft sogar eine Turtle eingesetzt und programmiert werden. Der Spieler setzt dann nicht mehr Stein auf Stein, um einen Turm zu bauen, sondern programmiert seinen Turtle-Roboter, der das erledigt. Der Spieler kann über einen visuellen Editor einfache Befehle ausprobieren, einfügen und kombinieren sowie Bedingungen oder Schleifen integrieren. Die Programme können abgespeichert und wiederverwendet werden.

Für Spieler ist dieser Moment der Automatisierung sehr motivierend, denn so werden die Spiel-Aktivitäten effizienter durchgeführt und Ergebnisse schneller sichtbar. Dadurch lernen die Kinder und Jugendlichen in ihrem Lieblingsspiel die Grundlagen des Programmierens fast von selbst.

Sie suchen Inspiration?
Kontaktieren Sie uns!

21CCC@fjs-ev.de
030 2938 1680
www.code-your-life.org

Sphero-Ball

Überblick

Thema	Robotertechnik und Gaming
Altersgruppe	ab 8 Jahre
Einsatzgebiet	Mathematik, Kunst, Informatik
Vorkenntnisse	Nicht zwingend notwendig
Kurzüberblick	Mit dem Sphero-Ball lernen die Kinder und Jugendlichen, wie sie einem Spielgerät Bewegungen, Farben und Kunststücke beibringen können. Sie verlassen die Ebene des Gaming und schauen hinter die Kulissen. Die Kinder und Jugendlichen schreiben Code, um den Ball zum Leuchten zu bringen, Kreise zu drehen oder über einen Parcours zu rollen.
Technologie	Der Sphero-Ball ist ein ferngesteuerter Spielball, der seine Farbe wechseln kann und mit dem Geschicklichkeitsübungen oder Rennen durchgeführt werden können. Er kann ein Rampe hochfahren, springen und sich drehen. Er ist äußerst robust und dabei nicht viel größer als ein Tennisball. Für den Sphero-Ball gibt es einige Apps mit denen man lustige Spiele spielen kann. Das Besondere daran ist, dass der Ball selbst programmiert werden kann.
Ressource	www.sphero.com

Im Rahmen von Code your Life wurde eine eigene Programmierumgebung für den Sphero-Ball entwickelt, der sogenannte Sphero-Coder. Bereits mithilfe einfacher Befehle kann man dem Sphero Leben einhauchen, ihm nach und nach das Rollen beibringen, ihn zum Tanzen animieren oder ihn sogar Hindernisse bewältigen lassen. Auch ohne Vorkenntnisse und Programmiererfahrung ist ein Erfolg sofort sichtbar. Recht schnell kommen Programmier-Logiken wie Variablen, Schleifen und Wenn-Dann-Bedingungen zur Anwendung und können praktisch ausprobiert werden.

Gerade mit Vorerfahrungen in der Programmierung der Turtle wird es hier spannend, denn der Sphero braucht andere Bezugsgrößen und Parameter, um sich im Raum zu bewegen. Im Gegensatz zum Mirobot bewegt er sich nicht nach Eingabe der Bezugsgröße Länge vorwärts, sondern reagiert auf Parameter von Geschwindigkeit und Richtung. Da der Ball kein Vorn und Hinten hat, ist die Richtung dann schon die erste größere Herausforderung, denen sich die Kinder und Jugendlichen stellen können.

Wurde das Prinzip erst einmal verstanden, können einzelne Bewegungsmodelle miteinander kombiniert werden, um ganze Choreographien zusammenzustellen oder einen Parcours abzufahren. Die Schwierigkeitsstufen können hier erhöht werden, wenn der Code für kompliziertere Formen, wie Spiralen gefunden werden soll.

Durch das integrierte LED-Licht, das verschiedene Farbskalen bereithält, macht das Ausprobieren der Codes im Dunkeln besonders viel Freude. In Kombination mit einer Video- oder Fotodokumentation können beeindruckende Videos und Langzeitbelichtungen á la Light-Painting für den Kunstunterricht erstellt werden.

Sie haben Interesse am Sphero-Coder?
Kontaktieren Sie uns!

21CCC@fjs-ev.de
030 2938 1680
www.code-your-life.org

Offline-Coding

Überblick

Thema	Programmieren ohne Computer
Altersgruppe	ab 6 Jahren
Einsatzgebiet	beliebig
Vorkenntnisse	Nicht notwendig
Kurzüberblick	Um grundlegende Programmierprinzipien zu lernen, braucht man zu Beginn nicht unbedingt einen Computer. Bei Code your Life ziehen wir auch einmal den Stecker und schauen uns die Prinzipien des Programmierens mit Papier und Bleistift, mit Bechern und Karteikarten oder sogar ohne jegliche Hilfsmittel, nur mit uns selbst als Medium, an.
Technologie	Die Ideen der Offline-Coding-Szenarien sind inspiriert von den Computer Science Unplugged Aktivitäten auf http://csunplugged.org/. Für die Durchführung werden keine Technologien, keine Computer und keine Software benötigt.

Offline-Coding eignet sich für Kinder und Jugendliche jeglicher Altersgruppe. Durch die Einbeziehung von Alltagssituationen können viele informatorische Algorithmen aus dem tatsächlichen Leben reflektiert werden und mit Programmierprinzipien in Verbindung gebracht werden. Die Faltanweisung eines Papierfliegers oder das Rezept für den leckeren Schokoladenkuchen sind beispielsweise ähnlich eines Computer-Skripts Abfolgen, die genau in der vorgegebenen Reihenfolge durchgeführt werden müssen, um das erwartete Ergebnis zu erzielen. Die Kinder lernen beim Offline-Coding, dass man beim Programmieren manchmal um die Ecke denken muss, aber mindestens genauso oft kreativ sein darf. Auch der ein oder andere Zaubertrick ist dabei und sie werden hautnah erkennen, dass Computerprogramme auch mal eine ganz andere Logik haben, als wir Menschen, z.B. wenn sie Dinge sortieren sollen.

Offline-Coding-Aktivitäten können und sollten immer wieder zwischen den Programmier-Einheiten am Computer stattfinden. Sie helfen den Kindern und Jugendlichen abstrakte Vorgänge zu verstehen, indem sie diese Vorgänge nachahmen. In den Code your Life Curricula wird dahingehend immer wieder empfohlen, selbst in die Rolle des Computerprogramms, z.B. der Turtle zu schlüpfen und die Kinder die Befehle ablaufen zu lassen. Plötzlich ist Programmieren keine abstrakte mathematische Herausforderung mehr, sondern eine Geschichte, die sich erleben lässt.

Eine schöne Offline Übung vor der Einführung von TouchDevelop und der Turtle ist die Heranführung der Kinder an Programmiersprachen, welche Sprache ein Programm verstehen kann und was eigentlich Befehle sind. In Zweier-Teams sitzen die Kinder jeweils Rücken an Rücken. Einer übernimmt die Rolle des Programmierers und einer die Rolle des Computerprogramms. Der Programmierer erhält ein Blatt Papier mit gezeichneten Symbolen und muss diese dem Computerprogramm beschreiben. Das Kind (Computerprogramm) zeichnet entsprechend der Anweisungen des Programmierers. Anschließend werden die Bilder verglichen und Rückschlüsse gezogen. In einem ersten Durchlauf sehen die Bilder meist sehr unterschiedlich aus. Den Kindern wird klar, dass sie genaue, klar definierte Befehle geben müssen, um nicht völlig unterschiedliche Interpretationen als gezeichnetes Bild zu bekommen. Mit diesem Wissen wenden die Kinder dann intuitiv Befehle an, die denen der Logo-Programmierung schon sehr ähnlich sind.

Haben Sie Lust auf Offline-Coding bekommen?
Kontaktieren Sie uns!

21CCC@fjs-ev.de
030 2938 1680
www.code-your-life.org

AntMe!

Überblick

Thema	Programmieren mit C# und Visual Studio
Altersgruppe	14-18 Jahre
Einsatzgebiet	Mathematik, Informatik
Vorkenntnisse	Erste Erfahrungen mit C# oder Visual Basic seitens des Lehrers sind hilfreich
Kurzüberblick	AntMe! ist ein kostenloses, spannendes Spiel mit dem junge Menschen einfach und mit viel Spaß lernen, wie man objektorientiert programmiert..
Technologie	Als Entwicklerumgebung bietet sich bei AntMe! Visual Studio an, das auch in der Softwareindustrie verwendet wird. In der Community Edition kann Visual Studio kostenfrei heruntergeladen werden. Die verwendeten Programmiersprachen sind C# und Visual Basic.
Ressource	www.antme.de

In AntMe! übernehmen die Schülerinnen und Schüler die Kontrolle über ein Ameisenvolk, das in einer virtuellen Welt auf Nahrungssuche geht oder gegen gegnerische Völker und Wanzen kämpft. Gespielt wird dabei aber nicht mit Maus und Tastatur, sondern mit Hilfe von Sourcecode, der das eigenständige Verhalten jeder einzelnen Ameise kontrolliert. Die Spieler bringen ihrem Ameisenvolk bei, wie sie verschiedene Herausforderungen meistern, wie z.B. Zucker und Äpfel zu sammeln, den Ameisenbau zu verteidigen oder sich gegen die natürlichen Feinde der Ameisen zu behaupten.

Einen guten Einstieg in AntMe! bieten die vorbereitenden Lerneinheiten, die durch das Spiel führen. Von dort stehen auch technischere Beschreibungen der Vorgänge zur Verfügung, die insbesondere Spielern mit Vorerfahrung hilfreich sind. Da AntMe! von Anfang an mit einer professionellen Programmiersprache wie C# arbeitet, kann durch das Spiel erlangtes Wissen sofort auf die Entwicklung eigener Programme übertragen werden.

AntMe!

In größeren Lerngruppen macht es den Kindern und Jugendlichen besonders viel Spaß, in kleinen Teams gegeneinander anzutreten. Dazu züchtet jedes Team sein eigenes Ameisenvolk und programmiert selbst gewählte Überlebensstrategien. Nach dieser intensiven Vorbereitungsphase werden die Ameisenvölker am Ende auf einem gemeinsamen Rechner „freigelassen". In einer Art Wettlauf zeigt sich, welche jungen Coder das stärkste Ameisenvolk programmiert haben, welche Ameisen scheitern und welche Strategien sich als die Besten heraus stellen. Denn eines ist klar: Es kann am Ende nur ein Ameisenvolk geben.

Sie haben Interesse am Programmieren mit AntMe?
Kontaktieren Sie uns!

21CCC@fjs-ev.de
030 2938 1680
www.code-your-life.org