

Informatik–Biber

*Lehrmittel für die informatische
Bildung an der Sekundarstufe I*

Kopiervorlagen



Arbeitsblatt 01: Programmieren mit Bausteinen

Fragestellung

Woraus besteht ein Programm?

Benötigte Materialien

- Aufgabe 1–3: Computer mit Internetzugang – www.code.org/learn
- Mehrere Plastikbecher, Stift und Papier
- Kopiervorlagen „Cup Stack Pack“ und „Anleitung Becherturm“

Aufgabe 1: Programmiere Anweisungen auf code.org

Öffne die Webseite: www.code.org

Klicke beim Programm „Code.org“ auf den „Start“-Button.

- Schau dir das Einstiegsvideo an. Zur Hilfe kannst du dir auch das Video im Text-Format anzeigen lassen. Löse die ersten 6 Aufgaben.
- Nenne zwei Anweisungs-Bausteine, die du verwendet hast. Was bewirken diese Bausteine?

Aufgabe 2: Programmiere Wiederholungen auf code.org

Öffne die Webseite: www.code.org

Klicke beim Programm „Code.org“ auf den „Start“-Button.

- Schau dir das Einstiegsvideo zu Aufgabe 7 an. Zur Hilfe kannst du dir auch das Video im Text-Format anzeigen lassen. Löse die Aufgaben 7 bis 10.
- Was ist der Unterschied zwischen einem „Wiederhole-Anzahl-Baustein“ und einem „Wiederhole-Bis-Baustein“?

Aufgabe 3: Programmiere Entscheidungen auf code.org

Öffne die Webseite: www.code.org

Klicke beim Programm „Code.org“ auf den „Start“-Button.

- Schau dir das Einstiegsvideo zu Aufgabe 14 an. Zur Hilfe kannst du dir auch das Video im Text-Format anzeigen lassen. Löse die Aufgaben 14 bis 20.
- Wofür brauchst du den „Wenn-Baustein“?
- Der „Wenn-Baustein“ lässt sich erweitern. Erkläre, wie und wann diese Erweiterung sinnvoll ist.

Aufgabe 4: Programmiere einen Mitschüler oder eine Mitschülerin

Das Ziel ist: Erstellt in einem Zweierteam ein Programm zum Bau eines Plastikbecherturms. Eine andere Zweiergruppe soll dann überprüfen, ob sie euren Turm mit eurem Programm bauen kann.

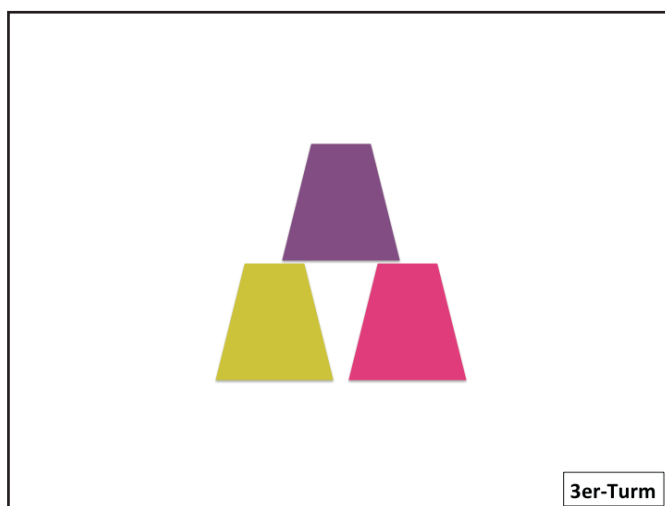
1. Such dir eine Mitschülerin oder einen Mitschüler. Schaut euch die beiden Kopiervorlagen „Liste aller Befehle/Schritte“ und „Schritte-Anleitung“ genau an, damit ihr die Befehle versteht, die ihr benutzen dürft.
2. Der Start ist für alle Türme immer gleich: Alle Becher sind ineinandergestellt und stehen links vor euch auf dem Tisch. Der Boden der Becher soll oben sein, die Öffnung unten. So wie in der Schritte-Anleitung auf Position „0 Schritte“.

Die Regeln

- Eure Hand ist wie eine Maschine, die sich nur gerade nach oben/unten bewegen kann, sowie gerade nach links und rechts.
- Beim Schritt „Becher aufnehmen“ hebst du den Becher immer bis auf die Höhe des obersten Bechers. Es gibt keinen Schritt „Becher eine Etage höher/tiefer setzen“. Der Becher in deiner Hand ist der oberste von allen.
- Ein Schritt (z.B. „Ein Schritt nach rechts“) ist immer die Hälfte der Becherbreite. D.h., damit ein Becher vom Stapel bis neben den Stapel gelegt werden kann, benötigst du 2-mal den Befehl „Ein Schritt nach rechts“. Schau dir die Schritte-Anleitung genau an!

- a) Spielt folgendes Beispiel selber durch: Könnt ihr mit dem Lösungsplan den 3er-Turm bauen?

So sollte euer Turm am Ende aussehen:



Ausgangslage: Stellt 3 Becher ineinander links von euch auf den Tisch (Becherstapel). Achtet darauf, dass der Boden der Becher oben liegt. Wir befinden uns in der Schritte-Anleitung nun auf Position „0 Schritte“.

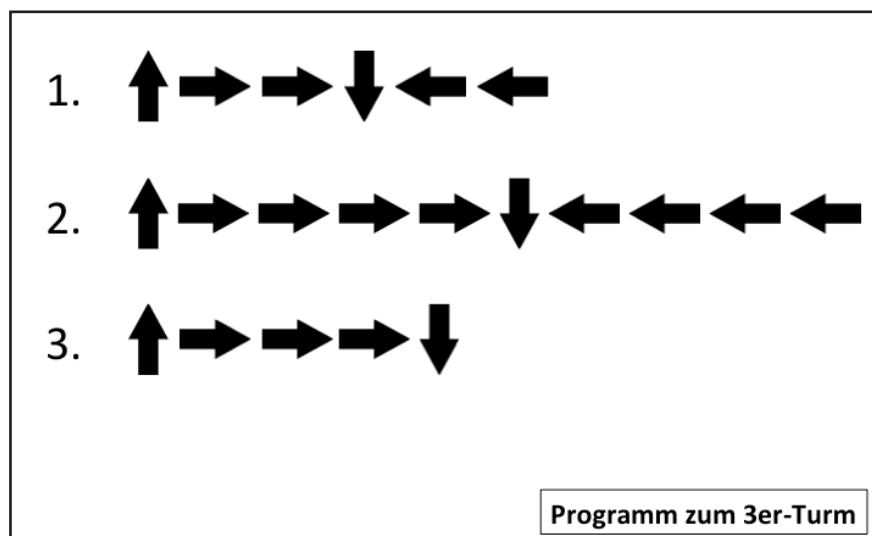
Eure Hand befolgt folgende Befehle:

1. Becher aufnehmen (denkt daran: Der Becher muss höher gehalten werden als der oberste Becher, den ihr auf eurem Turm habt!)
2. Ein Schritt nach rechts (denkt daran: Ein Schritt ist halb so gross wie die Breite eines Bechers!)
3. Ein Schritt nach rechts
4. Becher ablegen
5. Ein Schritt nach links (ihr habt keinen Becher mehr in der Hand)
6. Ein Schritt nach links

Ihr seid mit eurer Hand nun wieder an der Ausgangsposition bei eurem Becherstapel. Die Becher vor euch sollten aussehen wie in der Schritte-Anleitung auf Position „2 Schritte“.

Super! Ihr habt soeben die erste Zeile des Programms ausgeführt.

Versucht jetzt, die Zeile 2 und 3 auszuführen. Erhaltet ihr den Turm wie oben auf dem Bild?



- b) Erstellt ein eigenes Becher-Turm-Programm: Entscheidet euch, wie euer Turm am Schluss aussehen soll. Wählt eine der Zeichnungen von der Kopiervorlage, oder zeichnet selber einen Turm.
- c) Baut den Turm für euch auf, und schreibt mit den Befehlen jeden Schritt auf, den ihr mit dem Becher tut. Tipp: Eine Person führt die Bewegungen aus, die andere Person schreibt die Bewegung auf.
- d) Lasst euer Programm von einer anderen Zweiergruppe überprüfen. Sieht der Turm aus wie euer Turm?

Antworten zur Einstiegsfrage

Ein Programm besteht aus vielen einzelnen Anweisungen, die an einen Prozessor weitergegeben werden. Dieser führt die Anweisungen der Reihe nach aus. Um einem Computer verständliche Anweisungen mitzuteilen, benötigt man eine künstliche Sprache: die Programmiersprache. Es gibt viele unterschiedliche Programmiersprachen, die aber aus ähnlichen Bestandteilen (Programmelementen) bestehen. Zum Beispiel haben Englisch und Französisch Rechtschreib- und Grammatikregeln, die eingehalten werden müssen. Es sind aber zwei unterschiedliche Sprachen, weil sie unterschiedliche Wörter und einen anderen Satzbau (Syntax) benutzen. Ähnlich ist es mit den Programmiersprachen: Jede Programmiersprache hat ihre eigenen Regeln, die befolgt werden müssen. Werden diese Regeln missachtet, kann ein Computer die Anweisungen nicht mehr umsetzen, und es entstehen Fehler, häufig Fehlermeldungen. Wenn alle Anweisungen korrekt formuliert wurden, kann ein Programm ausgeführt werden, und erst wenn die Programmidee (der Algorithmus) auch richtig geplant war, dann macht das Programm das, was es sollte.

Arbeitsblatt 02: Event und Action mit Bausteinen

Fragestellung

Kann ein Event mehrere Actions auslösen? Und kann eine Action einen neuen Event auslösen?

Benötigte Materialien

- Computer mit Internetzugang – <http://scratch.mit.edu>
- Klicke auf „Entwickeln“, um in die Programmierumgebung zu kommen.

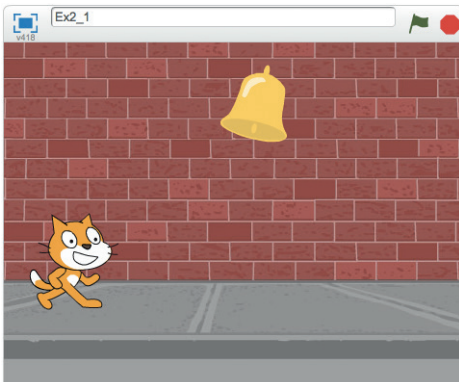
Aufgabe 1: Ein Event löst eine Action aus

Wandas Tram fährt in die Haltestelle ein. Nun vibriert ihr Smartphone und sie kann einsteigen. Wir wollen eine ähnliche Situation mit Scratch darstellen.

Programmiere Folgendes: Wenn die Glocke läutet (Vibraalarm vom Smartphone), soll die Katze loslaufen.

Ausgangslage: Wähle ein neues Bühnenbild, z.B. „brick wall1“. Positioniere die Figur „Katze“ an der linken Seite der Bühne auf dem Trottoir. Füge eine zweite Figur hinzu, und wähle die Glocke „Bell“ aus.

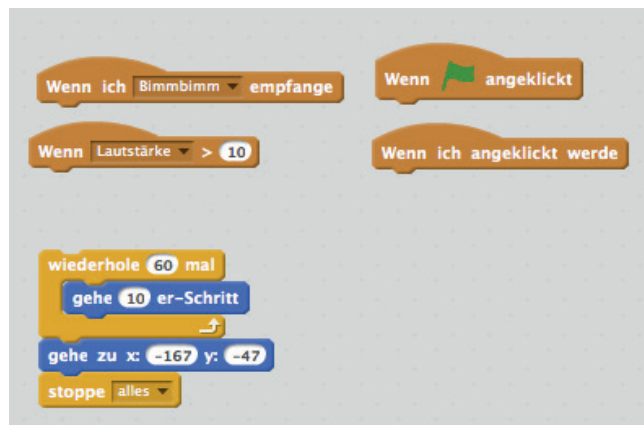
Das kann so aussehen:



Das sind die Anweisungen der Glocke:



Das ist das Skripte-Areal der Katze. Hier fehlt ein wichtiger Ereignis-Baustein (braun). Aber welcher der vier ist der richtige?



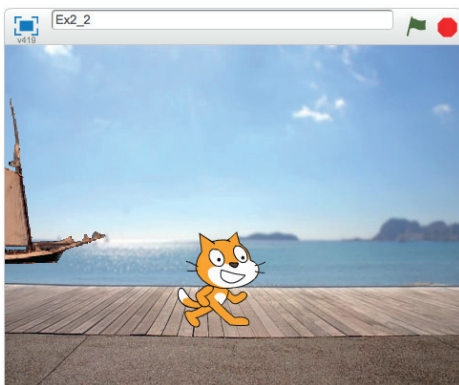
Aufgabe 2: Zwei Figuren kommunizieren miteinander

Solange Wandas Tram fährt, bleibt das Smartphone ruhig. Sobald das Tram an der richtigen Haltestelle stillsteht, fängt Wandas Smartphone an zu vibrieren und teilt mit, dass sie aussteigen muss.

Programmiere Folgendes: Solange das Schiff in Bewegung ist, bleibt die Katze stehen. Trifft das Schiff auf die Katze, bleibt das Schiff stehen, und die Katze läuft los.

Ausgangslage: Wähle ein neues Bühnenbild, z.B. „boardwalk“. Positioniere die Figur „Katze“ in der Mitte des Steges. Füge eine zweite Figur hinzu, und wähle „Sailboat“ aus.

Das kann so aussehen:



- a) Gehören diese Anweisungen zur Katze oder zum Schiff?
b) Gehören diese Anweisungen zur Katze oder zum Schiff?



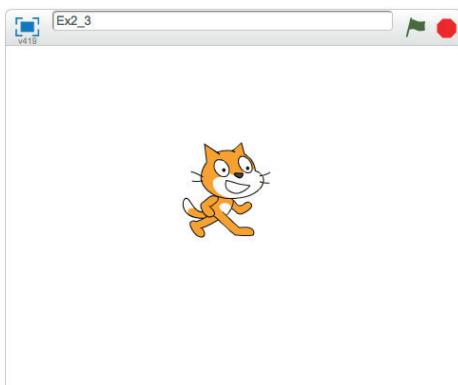
Aufgabe 3: Mensch – Maschine-Interaktion

Auf vielen Apps können mit Touchgesten Elemente herumgeschoben werden. Zum Beispiel verschiebt man eine App auf dem iPhone durch langes Gedrückthalten, bis es wackelt. Der Finger wird bewegt, und die App wandert wie magnetisch unter dem Finger mit.

Programmiere die Figur „Katze“ so, dass sie der Maus entlang folgt.

Ausgangslage: Positioniere die Figur „Katze“ in der Mitte der weissen Fläche. Starte mit dem Ereignis „Wenn ich angeklickt werde“.

Das kann so aussehen:



a) Welchen Bewegungs-Baustein brauchst du noch, damit die Katze dem Mauszeiger folgt?



b) Findest du eine Lösung, damit die Katze zum Mauszeiger „gleitet“?

Tipp: Die Katze muss die Koordinaten des Mauszeigers kennen. Benutze dazu die beiden Fühl-Bausteine „Maus x-Position“ und „Maus y-Position“.



Antworten zur Einstiegsfrage

Das Event-Action-Prinzip geht davon aus, dass ein Event (ein Ereignis) eine Handlung auslöst. Dies kann eine Kettenreaktion nach sich ziehen, sodass eine Action einen neuen Event auslöst und so eine zweite Action ausgeführt wird, u.s.w. Das könnte sich unendlich weiterziehen.

Jeder Event verlangt aber immer eine Action, welche aus mehreren Anweisungen bestehen kann, die nacheinander ausgeführt werden. Scratch bietet viele Beispiele für unterschiedliche Ereignisse an. Es gibt immer verschiedene Varianten, wie eine Handlung eingeführt werden kann. In der Programmierumgebung Scratch kann der Start eines Programms mit der grünen Flagge erfolgen oder aber auch über eine Taste oder gar über eine Nachricht, die empfangen wird. Ein Programm kann somit auf unterschiedliche Arten aufgerufen werden. Meist gibt es verschiedene Lösungen für die korrekte und richtige Ausführung eines geplanten Programms. Jede/r Programmierer/in gibt so seine/ihre eigene Handschrift in ein Programm.

Arbeitsblatt 03: Rund um die App

Fragestellung

Wann ist eine App eine „barrierefreie“ App? Was heisst „barrierefrei“ eigentlich?

Benötigte Materialien

- Kopiervorlage Bausatz

Aufgabe 1: Was soll die App können?

a) Wir wollen Wandas App weiterentwickeln. Da sie nicht mehr so gut sieht, braucht sie auch Apps, die ihr helfen, besser zu sehen. Sie hat Schwierigkeiten, Farben zu unterscheiden, kleine Schriften kann sie nicht mehr lesen (z.B. Zubereitungsanleitung auf einer Packung), und gewisse Dinge erkennt sie nicht mehr von Auge, und jemand muss ihr sagen, was es ist. Wer ausser Wanda kann diese App ebenfalls brauchen? Fallen dir noch mehr Ideen ein, wie man Menschen mit einer Sehbehinderung auch noch unterstützen könnte?

Formuliere drei verschiedene Nutzerinnengeschichten. D.h. beschreibe genau, in welchen Situationen diese App eingesetzt wird und von wem. Was muss diese App können?

- ...
- ...
- ...

Eine Nutzerinnengeschichte könnte zum Beispiel so lauten:

„Wanda ist körperlich behindert. Sie hört nicht mehr so gut, und Farben kann sie nicht mehr sicher unterscheiden. Die Farberkennung kann sie einsetzen, um die Farbe einer ihrer Balkonblumen zu bestimmen.“

Aufgabe 2: Wie soll die App aussehen?

Erstellt in einer Zweier- oder Dreiergruppe mithilfe der Bausatz-Elemente ein eigenes App-Design für Wandas-Farberkennungs-App. Überlegt euch dabei Folgendes:

- Was muss die Nutzerin auf der App sehen können?
- Welche Informationen können abgerufen werden? Und wie?
- Wer sind eure Zielpersonen?
- Achtet darauf, dass die Bedienbarkeit einfach ist.

Ihr könnt mehrere Bildschirm-Bausätze erstellen. Sobald man auf einen Button klickt, wird man ja meistens auf eine weitere Seite weitergeleitet.

Aufgabe 3: Funktioniert die App?

Legt euer App-Design einer anderen Gruppe vor, und testet so eure App. Lasst die anderen beschreiben, was diese sehen und welche Funktionen sie sich darunter vorstellen.

- Erkennen die anderen, wo sie die Funktionen starten können?
- Erkennen die anderen, wofür welcher Button genutzt wird?
- Könnt ihr noch etwas verbessern und anpassen?

Antworten zu den Einstiegsfragen

„Barrierefrei“ heisst, dass etwas für alle Menschen mit oder ohne körperliche Behinderung oder Einschränkung zugänglich ist. Der Begriff kommt ursprünglich aus dem Bauwesen, wo Gebäude z.B. für Rollstuhlfahrende zugänglich gemacht werden müssen. Diese Gebäude sind (wenn sie alles beachten) ohne Hindernisse, d.h. ohne Barrieren (= barrierefrei). Der Begriff wird aber auch für Webseiten oder für Apps benutzt. Wenn eine Webseite zum Beispiel auch für Blinde frei zugänglich ist, ist sie „barrierefrei“. Blinde benutzen meistens ein Programm zum Vorlesen einer Webseite, einen sogenannten Screenreader. Wenn die Webseite barrierefrei aufgebaut ist, können Blinde mithilfe des Vorleser-Programms auch Bilder erkennen und wissen, wie die Webseite aufgebaut ist. Auch Apps sollten barrierefrei sein, so dass jeder Mensch sie benutzen kann. Die App von Tante Wanda zum Beispiel ist barrierefrei programmiert, damit sie die App bedienen kann, auch wenn sie nicht mehr so gut sieht.