

Aufgabentyp 2 - «MODIFY»

Modify

Lernende lernen programmieren mittels bereits bestehender Programme - nun, im zweiten Schritt in dem diese geändert werden. Die Lernenden sind "Editoren". Meist werden anfänglich triviale Aspekte wie Farben, Formen, Schrittweiten, etc. geändert; später auch ganze Abläufe, Elemente, Interaktionen, usw.

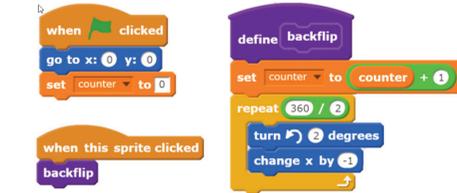
(Mögliche) Aufgabentypen:

- Programme debuggen (Code zu Text und Code). Fehler werden gesucht und das Programm korrigiert.
- Programme erweitern (Code zu Text und Code). Ausnahmen werden getestet und das Programm so optimiert, dass es in jedem Fall ausgeführt werden kann.

Leitfragen: Wo ändere ich es? Wie ändere ich es?

Hilfsmittel: Ermutigen, auszuprobieren. Neu starten, wenn das Programm „kaputt geht“. Motivation. Wertschätzung ermöglichen. Fortschritte betonen. Misserfolge ignorieren. Analogie Skateboarden: umfallen und wieder aufstehen.

Beispiel «MODIFY»



Führe das Programm in Scratch aus.

- ? Was passiert?
- ? Was passiert, wenn Du den Ball anklickst?
- ? Ändere die Zahlen 0 und 0 im blauen Block auf 50 und 100 ab. Was passiert?
- ? Was bedeuten die Zahlen?
- ? Was bedeutet also x? Und was bedeutet y?
- ? Verändere einmal die Zahl 2 im Block rechts „turn 2 degrees“ auf 1 ab. Was passiert? Du kannst nun eine ANDERE Zahl anpassen, damit das gleiche passiert wie am Anfang. Findest Du heraus, welche Zahl und auf welchen Wert Du sie setzen musst?
- ? Was bewirkt das „change x by -1“? Probiere Werte aus.
- ? Kannst Du einbauen, dass beim Klicken auf den Ball ein Geräusch ertönt?

Aufgabentyp 3 - «CREATE»

Create

Lernende lernen programmieren anfänglich mittels bereits bestehender Programme - in dem diese programmatisch erweitert werden (nicht nur Parameter ändern). Später auch, in dem Vorgaben gemacht werden, die die Lernenden mittels eigenen Programmen umsetzen/erfüllen. Sie sind "Planer" und "Produzenten".

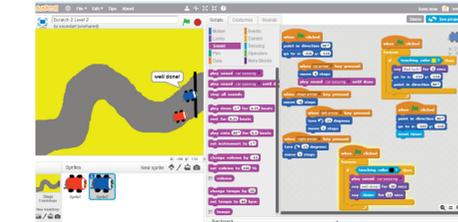
(Mögliche) Aufgabentypen:

- Programme erweitern, so dass neue Elemente (wie z.B. Sprites, Spielfiguren, etc.) oder auch Funktionen, Interaktionen, Features usw. hinzukommen. Die Vorgaben können in Bild-, Video- oder Textform vorliegen oder auch z.B. in Form von User-Stories (Kundenwünsche).
- Programmieren nach bildlichen Vorgaben (Bild zu Code). Das Produkt soll nach einer Bildergeschichte nachprogrammiert werden.
- Programmieren nach Vorgaben als Video (Video zu Code). Das Endprodukt wird als Video präsentiert. Die Schülerinnen und Schüler überlegen sich, wie sie die Vorgaben erfüllen können und erstellen das Programm dazu (Scratch-Challenges in der Klasse, über Klassen hinweg).
- Programmieren nach textlichen Vorgaben (Text zu Code). Die Schülerinnen und Schüler programmieren nach einer Text-Vorgabe. Bonus-Ziele können zur Differenzierung eingesetzt werden.
- Programmieren eigener Ideen nach textlichen Vorgaben (Text zu Code). Eine Grundidee wird vorgegeben, welche mit individuellen Umsetzungen erreicht werden kann.
- Programmieren eigener Ideen ohne Vorgaben. Eigene Ideen werden umgesetzt. Plots für Software entwickelt. Sprites und Hintergründe kreiert und ganze Spiele erfunden und umgesetzt.

Leitfragen: Was könnte ich noch einbauen? Wie bauen ich es ein? Wie stelle ich sicher, dass es in jedem Fall funktioniert?

Hilfsmittel: Ein Abgleich mit den Vorgaben ist hilfreich, auch regelmäßiges Feedback, Präsentationen, oder Paar-Programmierung. Alle Schwierigkeitsgrade ermöglichen. Nicht auf Scaffolding und formative Feedbacks verzichten

Beispiel «CREATE»



Führe das Programm in Scratch aus.

- ? Verändere die Spielmusik und die Geräusche, so dass sie Dir noch besser gefallen. Könnten alle Geräusche zu einem gleichen Thema passen (z.B. Safari)?
- ? Gestalte den Hintergrund der Rennbahn wie in der Arktis.
- ? Kannst Du ein drittes (grünes) Auto einbauen?
- ? Kannst Du ein Auto so ändern, dass es ein Kamel ist und schneller reiten kann, als das andere Auto?
- ? Baue ein Zeitanzeig, die nach dem Start die Sekunden zählt, wie lange die Autos schon fahren. Die Zeit soll am Ziel stoppen.
- ? Hast Du eine Idee, wie man das Spiel noch spannender gestalten kann? Welche zusätzlichen Elemente könntest Du noch einbauen?
- ? Lass Dir eine Aufgabe von Deinem Banknachbarn stellen und setze sie um. Bist Du gespannt, ob Du es in einer Stunde schaffst? Stelle dafür einen Timer.

Wie werden Hausaufgaben geplant und reflektiert?

Prozessschritt

Analysieren



Entscheiden

Entwerfen

Realisieren

Reflektieren

Leitfragen für die Lehrperson

- Fachliche und überfachliche Voraussetzungen**
 Was wissen die Lernenden bereits zum Thema?
 Welchen Lebensweltbezug haben die Lernenden?
 Welche Kompetenzstufen sollen erreicht werden?
 Welche Inhalte und Zusammenhänge können von den Lernenden selbstständig erarbeitet werden?
 Welche Arbeitsmethoden kennen die Lernenden bereits?
 Wie verhalten sich die Lernenden bei Unsicherheiten zu den Hausaufgaben während der Realisierung?
 Welche Hilfestellungen sind nötig?
- Personale und soziale Voraussetzungen**
 Welche entwicklungsbedingten Voraussetzungen müssen bedacht werden?
 Welche Werte und Haltungen prägen den Lernprozess?
 In welcher Umgebung arbeiten die Lernenden?
- Strukturelle Voraussetzungen**
 Welche Infrastruktur steht den Lernenden zur Verfügung?
 Welche Regeln müssen beachtet werden?
 Wie viel Zeit kann für Hausaufgaben eingesetzt werden?
 Welche Zeitfenster können genutzt werden?
- Welche Ziele sollen die Lernenden erreichen?
 Welche Differenzierung der Lernziele wird bei den Lernenden vorgenommen?
 Sind diese Ziele mit dem Lehrplan konform?
 Sind diese Ziele eine Herausforderung und gleichzeitig realistisch?
- Wo befinden sich die Lernenden im Lernprozess?
 Welche Lernaufgaben eignen sich für die Hausaufgabe?
 Welche Methoden eignen sich für die Hausaufgabe?
 Welche handlungsorientierten und/oder spielerischen Formen eignen sich für die Hausaufgabe?
 Auf welche Weise werden die Lernergebnisse überprüft?
- Wie werden die Hausaufgaben begleitet?
 Auf welche Weise korrigiert / bespricht die Lehrperson die Hausaufgaben?
 Was hat gut funktioniert? Warum?
 Was hat nicht funktioniert? Warum?
 Welche Massnahmen unterstützen die Hausaufgabenpraxis?
- Was hat gut funktioniert? Warum?
 Was hat nicht funktioniert? Warum?
 Welche Massnahmen unterstützen die Hausaufgabenpraxis?

Wo stehen Lernende im Lernprozess? Welche Lernaufgaben eignen sich? Welche Fragen helfen sich zu orientieren?

PADUA-Modell

Problemstellung

- Konfrontationsaufgaben
- Wo finde ich Aspekte des Themas in meinem Leben?
- Warum ist das Thema für mich interessant/wichtig/relevant?
- Was weiss ich schon?
- Was möchte ich noch wissen / können?

Aufbau

- Erarbeitungsaufgaben
- Wie funktioniert das? Welche Konzepte stehen dahinter?
- Welche Begriffe brauche ich, um das Thema zu erklären?

Durcharbeiten

- Vertiefungsaufgaben
- Welche Gemeinsamkeit und Unterschiede bestehen zwischen verschiedenen Situationen?
- Welche Verhaltensregeln lassen sich ableiten?

Lieben

- Übungsaufgaben
- Was muss ich machen, damit ein Teilaspekt funktioniert?
- Welche Begriffe muss ich für eine Erklärung verwenden?

Anwenden und Überprüfen

- Transfer- und Synthesaufgaben
- Was muss ich genau machen, damit das Ganze funktioniert?

- Formative und summative Beurteilungsaufgaben
- Wissen: Was weiss ich (noch nicht)?
- Anwendung: Was kann ich (noch nicht)?
- Vorgehen: Wie bin ich vorgegangen?
- Was eignet sich (nicht) für mich?
- Warum (nicht)?



Impressum

Autoren
Marcel Mösch
Raffael Meier



Dieses Faltblatt entstand als Leistungsnachweis im Modul Wissenschaftstransfer im Rahmen des Studiengangs Master Fachdidaktik Medien & Informatik an der Pädagogischen Hochschule Schwyz im Juli 2021.
Dozent: Dr. Martin Hermida

Praxisleitfaden
Ein Wissenschaftstransfer-Projekt
von Marcel Mösch & Raffael Meier

HAUSAUFGABEN INFORMATIK PROGRAMMIEREN

www.informatikhausaufgaben.ch

© 2021 by Marcel Mösch und Raffael Meier
Bildquellen: pixabay.com
Für verlinkte Inhalte wird jegliche Haftung abgelehnt.
Alle Rechte, insbesondere der Publikation und Vervielfältigung, (erst einmal) vorbehalten.