

Mathematik mit Scratch ab Jahrgangsstufe 3: Orientierung im Raum

Inhalt

1. Fachanforderungen Mathematik
2. Gestalten für und Programmieren mit Scratch
3. Einfache Beispiele zur Kernidee „Orientieren im Raum“

Fachanforderungen

In den Fachanforderungen Mathematik wird für den „Inhaltsbereich Raum und Form“ gefordert, dass im „Unterricht [...] in einem Spiralcurriculum durch treffende Unterrichtsbeispiele die Kernideen Orientierung im Raum, ebene Figuren, räumliche Objekte, Symmetrie und Zeichnen“ behandelt werden. Die zu entwickelnde „Kompetenz“ ist das „räumliche Vorstellungsvermögen“. Das heißt, die Schülerinnen und Schüler:

- „orientieren sich mit Hilfe von Plänen.“
- „erkennen und beschreiben Wege- und Lagebeziehungen anhand von Plänen.“
- „bewegen Objekte in ihrer Vorstellung und beschreiben den Vorgang.“
- „erkennen, beschreiben und nutzen den Zusammenhang zwischen dreidimensionalen Objekten und ihren zweidimensionalen Darstellungen.“

Unter „Vorgaben und Hinweise“ wird ausgeführt, dass „neben den handwerklich-praktischen Aspekten [...] ab der 3. Jahrgangsstufe gedanklich-theoretische Aspekte eine stärkere Rolle“ einnehmen.

Gestalten für und Programmieren mit Scratch

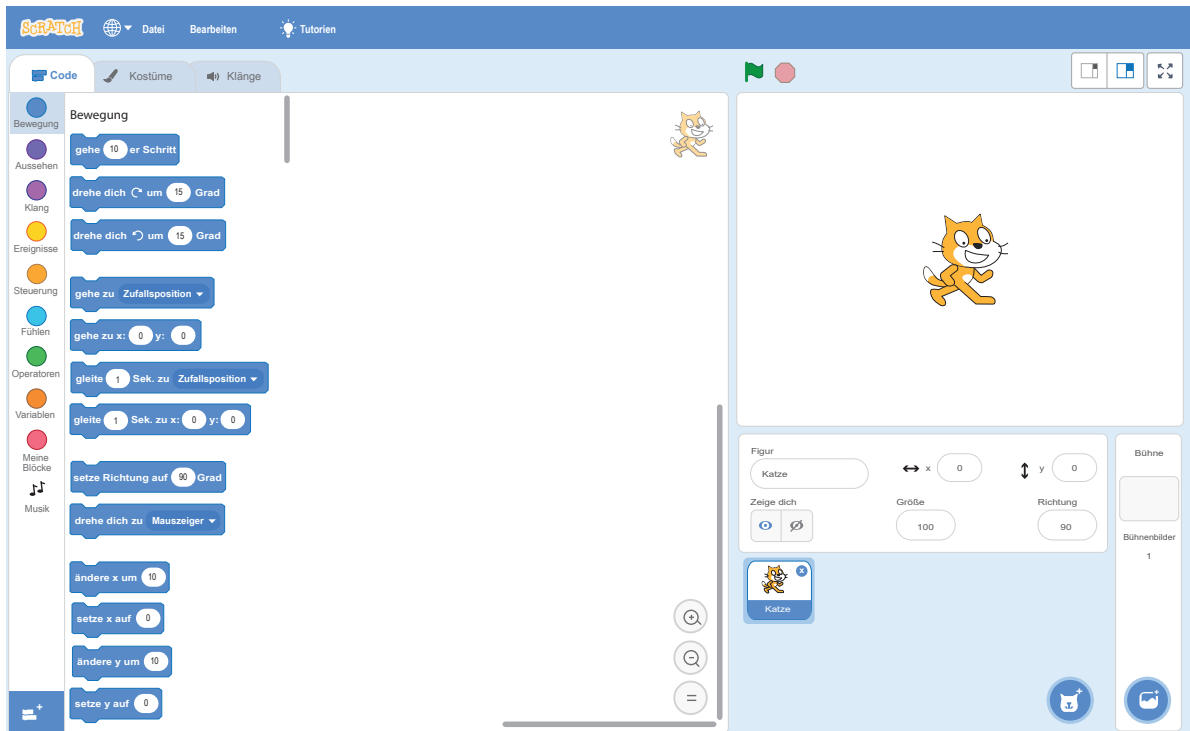
Wie solche Aspekte spielerisch mit Scratch erarbeitet und vertieft werden können, soll im Folgenden verdeutlicht werden.

„Scratch“ übersetzt man am besten mit „zusammenkratzen“. Gemeint sind selbst gezeichnete oder aus den programmeigenen Bibliotheken ausgewählte Bilder, die entweder „Bühne“ oder bewegliche Objekte, sogenannte „Figuren“ sind.

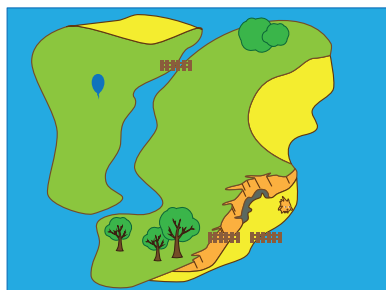
Nach dem Start des Programms wird auf der rechten Seite eine stilisierte Katze gezeigt. Auf der linken Seite finden sich blaue „Codeblöcke“, mit denen Bewegungsmuster ausgelöst werden können. Beispielsweise bewegt ein Klick auf den Block „gehe 10er Schritt“ die Katze ein Stück weit nach rechts.

Die Beschriftung der Blöcke im Zusammenspiel mit der Klick-Erprobung machen das Programmieren leicht.





Der Zugang zu den vorgehaltenen Bühnenbildern und Figuren bzw. den Zeichfunktionen verbirgt sich hinter den beiden runden Schaltflächen auf der Programmoberfläche rechts. Ein Klick darauf öffnet ein selbsterklärendes Auswahlménü wie in der Abbildung rechts.



Gobo



Balloon1

Für die Aufgabe wurde als „Bühne“ eine Insel-Landschaft gezeichnet, auf der „Gobo“ auf dem kürzesten Weg einen „Ballon“ (beides Figuren aus der Bibliothek) einsammeln soll.

Die Größe der beiden Figuren lässt sich stufenlos der Bühne anpassen. Die Katze wird über einen Klick von der Bühne gelöscht.

Mit gedrückter linker Maustaste können Gobo und der Ballon an beliebigen Stellen der Insel-Landschaft platziert werden.

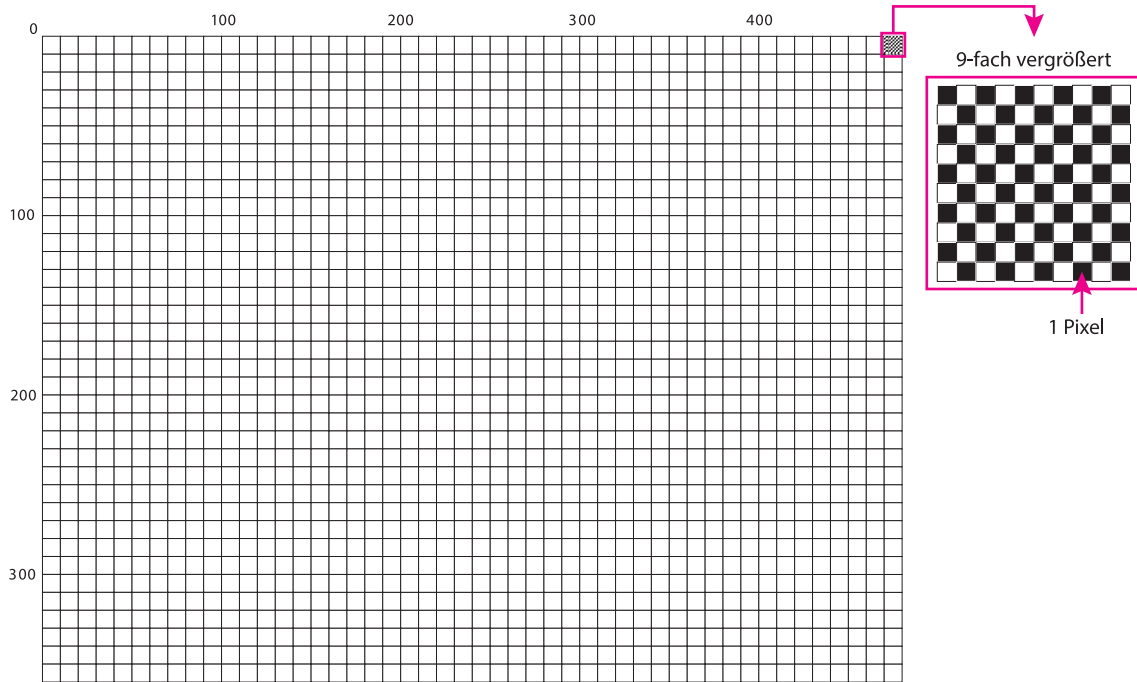
Bühnenbild wählen



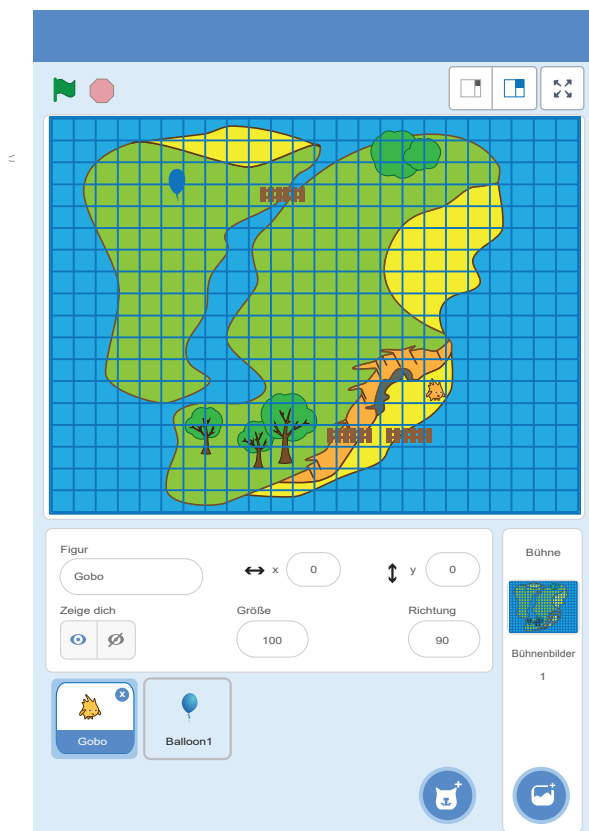
Um den Gobo zum Ballon führen zu können, muss man wissen, dass die Bühne eine Größe von von 960x720 Pixeln hat und die Figuren pixelgenau gesteuert werden können: Ein 10-er Schritt entspricht also eine Strecke von 10 Pixeln.

Die nachfolgende Zeichnung soll verdeutlichen, wie klein ein Pixel ist und wie genau deshalb eine Figur über einen Bildschirm gesteuert werden kann.

Damit die einzelnen Pixel auf der Seite noch erkennbar sind, wird die In Scratch 3.0 vorgegebene Größe der Bühne von 960x720 Pixel auf 480x360 halbiert und die Fläche in 48x36 Felder aufgerastert. In jedem dieser Felder muss man sich 10x10 = 100 Pixel denken, was einer Gesamtzahl von 172.800 Pixeln entspricht.



Legt man ein passend dimensioniertes Netz über die Insellandschaft, können Lage des Ballons und der Weg dahin auf einfache Weise bestimmt werden. Die einzelnen Felder müssen zur Größe der Figuren passen und repräsentieren eine gerade Pixelzahl. Ein Netz mit $24 \times 18 = 432$ Felder erscheint geeignet. Jedes Feld vereint damit $20 \times 20 = 400$ Pixel, was 20-er Schritte nötig macht, um sie zu durchqueren.



Sind Bühnenbild und Figuren geladen und platziert, zeigt sich das in der Scratch-Programmierungsumgebung wie in der Abbildung links.

Die Figur „Gobo“ und der Reiter „Code“ müssen angewählt sein, damit das Programm für den Weg zum Ballon festgelegt werden kann.

Da die Bühne vergrößert werden kann und der Programmcode dann von ihr verdeckt wird, soll die Ausführung des Codes mit Klick auf die grüne Flagge initiiert werden können.

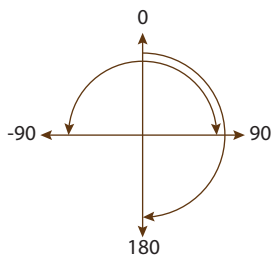
Um jeden Schritt des „Gobo“ nachvollziehen zu können, werden Pausen eingearbeitet.

Erreicht die Figur den Ballon, wird nach kurzer Wartezeit die Figur auf die Ausgangsstellung zurückgesetzt und damit ein Neustart ermöglicht.

Der erste Block ermöglicht es, bei einer Vergrößerung der Bühne das Programm zu starten.

Die Figur nicht zu drehen verhindert, dass sie bei Richtungsänderungen aus dem Netz herausfällt.

Die Abbildung unten erklärt die Gradzahlen.



Die Schrittweite ist dem Rasternetz angepasst. Bei längeren Wegstrecken wird auch mal über mehrere Felder gesprungen.

Die Pausen ermöglichen es dem Betrachter, den Weg zu verfolgen.

Ganz zum Schluss soll die Figur für 10 Sekunden am Ziel verharren und dann an den Ausgangspunkt zurückkehren.

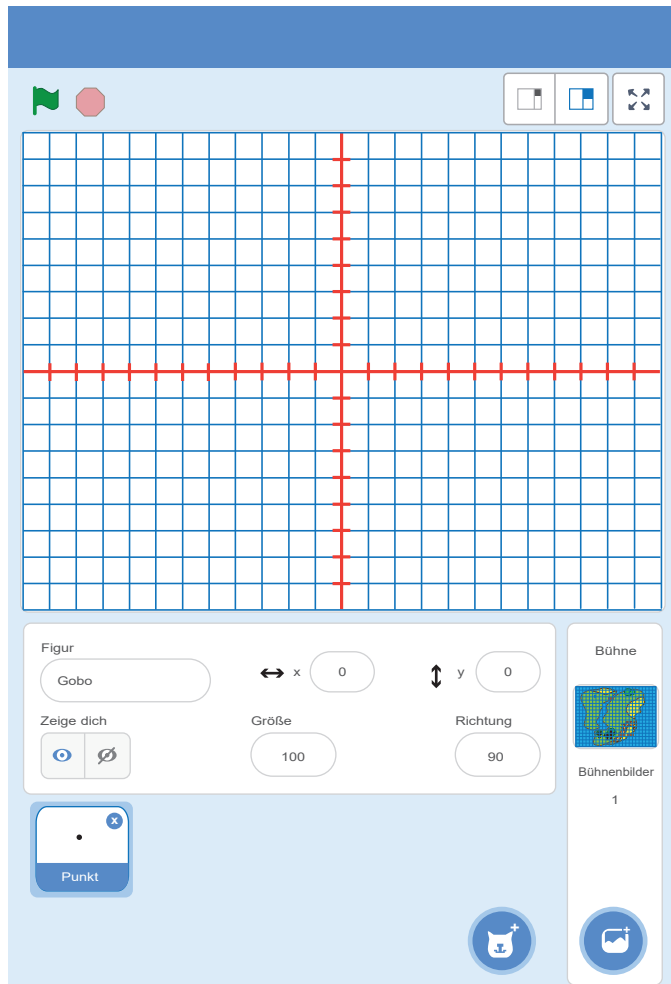
Das Programm kann so beliebig oft gestartet werden.



Einfache Beispiele zur Kernidee „Orientieren im Raum“

Das Koordinatenkreuz

Scratch bringt 3 für diese Zwecke halbwegs geeignete Grafiken mit: Xy-grid, Xy-grid-20px und Xy-grid-30px. Die erstgenannte Grafik ist ein Koordinatenkreuz mit 100-er Einteilung. Die beiden anderen sind Netze mit einem Rastermaß von 20x20 und 30x30 Pixeln. Wer mehr will, muss sich geeignete Vorlagen selbst erstellen. Die im Folgenden verwendeten können [hier](#) heruntergeladen werden.



Als „Figur“ dient eine kleine Kreisfläche (Punkt), die „Bühne“ ist ein 20er-Rasternetz mit einem Koordinatenkreuz. Die Aufgaben dazu lauten:

1. Steuere in jeweils einem Schritt folgende Punkte an:
a. Auf der x-Achse: -40; 20
b. Auf der y-Achse: 120; -60

2. Bewege den Punkt auf dem Koordinatenkreuz in alle vier Richtungen

3. Ermittle den Abstand zwischen Einteilungsstrichen

4. Bewege den Punkt in zusammenhängenden Schritten zu folgenden Koordinaten:

$$x=-80 / y=120$$

$$x=-80 / y=-120$$

$$x=80 / y=-120$$

$$x=80 / y=120$$

Starte bei $x=80 / y=120$. Die Bewegung und die zurückgelegten Strecken zwischen den angegebenen Koordinaten sollen erkennbar sein. Wie heißt die so gezeichnete Figur?

5. Programmiere das „Haus des Nikolaus“ in Rot auf weißem Hintergrund. Nach einer kurzen Verweilzeit soll die Zeichnung gelöscht und der Ausgangspunkt wieder eingenommen werden.

Zu1



Die Zahlen in den weißen Feldern können angeklickt und dann durch gewünschten Werte ersetzt werden.

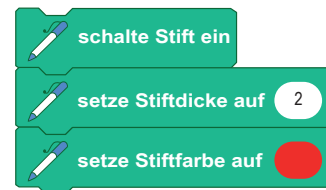
Zu 2 und 3

setze Richtung auf -90 Grad Ein Klick auf die Zahl ermöglicht die Richtungswahl: -90 nach links, 90 nach rechts, 0 nach oben, 180 nach unten.

gehe 10 er Schritt Die Schrittweite lässt sich durch Verändern der Zahl im weißen Feld einstellen.

Zu 4

Für die Programmierung wird ein Malstift benötigt. Dieser kann mit den dazugehörigen Programmblöcken auf dem Startbildschirm durch Anklicken des blauen Feldes ganz links unten gefunden und eingeblendet werden. Die Stiftfarbe kann durch einen Klick auf das Farbfeld im Programmblock geändert werden. Entsprechendes gilt für die Stiftdicke.



Zu 5

Ein günstiger Startpunkt für die Zeichnung ist beispielsweise $x=-140 / y=-150$. Für das Zurücksetzen der Zeichnung auf den Ausgangszustand erscheinen folgende Böcke hilfreich:



Rösselsprung

Im Schachspiel kann ein Pferdchen, wie in der Abbildung rechts gezeigt, in vier Richtungen bewegt werden und zwar immer zwei Felder vor (links, rechts, oben, unten) und eins zur Seite (links, rechts). Insgesamt kann das Pferdchen mit einem Zug, dem sogenannten „Rösselsprung“, acht verschiedene Felder erreichen.

Zur Programmierung eines Rösselsprungs müssen Richtungen und Schrittweiten festgelegt werden. In dem Code-Beispiel rechts wird das Pferdchen nach oben und links bewegt.

Eine passende Aufgabe dazu könnten sein, ein bestimmtes Feld im Koordinatensystem mit möglichst wenigen Zügen zu erreichen.

