

# Scratch @yourschool

## Ein eigenes Spiel in 3h - genereller Ablauf

### Was passiert im Kurs?

- Der Kurs beginnt mit der Begrüßung im Plenum. Ein erfahrener Inspirer (Haupt Inspirer) übernimmt die Einführung in das Thema. Danach werden die Teilnehmende in Kleingruppen eingeteilt. Jeder Inspirer ist während des Kurses für etwa 4 bis 5 Kids verantwortlich.
- Online: Die eingeteilten Teams werden in Breakout-Rooms (bei Zoom) eingeteilt. Wenn ihr in den Breakout Rooms Fragen habt, ruft gerne über Zoom um Hilfe. Der Haupt-Inspirer wird dann zu euch in den Breakout-Room kommen.
- In Präsenz: Die eingeteilten Teams setzen sich an Gruppentische, an dem je ein Inspirer sitzt. Fragen können direkt dem Inspirer, in der entsprechenden Gruppe oder dem / der Hauptinspirer, gestellt werden.
- Vorstellungsrunde in den Gruppen. Stellt euch einmal vor und erklärt gerne, wie ihr zu der IT gekommen seid. Lasst die Kids sich vorstellen, inklusive Programmiererfahrung, Erwartungen und Motivation.
- Jetzt geht's ans Eingemachte: Programmieren! Mehr Informationen zu den Einheiten findet ihr unten.
- Versucht nach etwa der Halbzeit eine Pause von 30 Minuten einzulegen.
- Der Abschluss findet wieder im Plenum mit der gesamten Klasse statt. Hier stellt sich jeder Inspirer einmal kurz vor und erklärt, warum er/sie für IT brennt.

P.S.: Vergesst nicht, eure Pausen zu machen :)

## Hinweise zum Konzept - Allgemeine Informationen

### Lernziele - Pädagogisches Konzept

- Es geht darum, die Kids zu begeistern. Unser Ziel ist es, dass sie sich nach dem vierstündigem Kurs noch hinsetzen und weiter recherchieren und programmieren. Wie das bei euch konkret aussieht, ist euch überlassen. Ihr merkt, die Kids haben voll Lust mit euch eine halbe Stunde über Roboter zu reden? Dann macht das gerne :)
- Versucht eher, den Kids grundlegende Konzepte beizubringen ("Wie debugge ich ein Programm?"), als Ihnen diese eine Eigenheit lang und breit zu erklären. Zeigt/erklärt ihnen auch gerne, wo das, was sie gerade lernen, im echten Leben zum Einsatz kommt.
- Lasst die Kids so viele Schritte wie möglich selbst gehen. Sie werden so am meisten lernen, und sind hoffentlich auch in der Lage, an das Gelernte nach Ende des Kurses anknüpfen zu können.
- Lasst sie dabei auch scheitern. Im Optimalfall versucht ihr dann mit ihnen gemeinsam zu ergründen, wieso ihr Ansatz nicht funktioniert hat - dadurch müssen sie ihre Gedanken kommunizieren, und lernen so mehr.
- Dieses Konzept ist unser Vorschlag, wie man Kinder und Jugendliche begeistern könnte. Seht es also mehr als groben Leitfaden und weniger als festen Lehrplan.

## Scratch - Online Editor

- Wir arbeiten während des Kurses mit dem Online-Editor von Scratch -> <https://scratch.mit.edu/projects/editor>
- Scratch ist eine graphische Programmiersprache vom MIT, die es besonders einfach macht eigene kleine Spiele zu entwickeln.
- Der Online-Editor funktioniert ohne Account, wir können also sofort loslegen. Wenn man trotzdem Projekte speichern möchte, muss man sich entweder einen Account erstellen oder über „Datei > Auf deinem Computer speichern“ die Datei speichern.
- Wir gehen den Editor im Briefing einmal kurz mit euch durch.

## Vorlage Beispielprojekte

Wir haben für den Kurs ein (mögliches) Projekt fertiggestellt, das ihr euch anschauen, herunterladen und als Anhaltspunkt verwenden könnt:

<https://scratch.mit.edu/projects/513952535>

Ein paar denkbare Erweiterungen gibt es auch:

<https://scratch.mit.edu/projects/514884972>

<https://scratch.mit.edu/projects/514887282>

Wenn ihr wollt, könnt ihr selbst im Voraus Erweiterungen implementieren, um sie den Teilnehmer:innen als Inspiration zu geben oder einige Features gemeinsam zu implementieren (Details: siehe Erweiterungen). Dafür bietet es sich an das Beispielprojekt zu kopieren damit die Codebasis die gleiche ist wie hier im Kurs beschrieben. Falls ihr diese Erweiterungen später auch anderen Inspirern zur Verfügung stellen wollt, könnt ihr uns das Projekt gerne an [dataclub-lueneburg@leuphana.de](mailto:dataclub-lueneburg@leuphana.de) schicken!

## Kurskonzept Was passiert im Kurs

### Begrüßung und generelle Motivation 15 Min - Plenum - Haupt Inspirer

Diesen Teil übernehmen wir. Er steht nur da, damit du einen besseren Überblick bekommst und verstehen kannst, wie wir gerne mit den Kindern zusammenarbeiten würden.

- Wer sind wir und was wünschen wir uns? (an die Kinder gerichtet)
  - Wir wollen, dass ihr alle etwas mitnehmen könnt. Uns ist es lieber, dass ihr Zusammenhänge verstanden habt, aber dafür weniger. (Wird trotzdem vorkommen, dass euch Dinge überfordern, aber das ist normal. Behaltet das trotzdem im Hinterkopf)
  - Vordergründig bringen wir euch heute Programmieren bei, aber eigentlich geht es uns darum, euch mit Menschen zusammenzubringen, die gemeinsam mit euch lernen und sich (genauso wie ihr) weiterentwickeln wollen. Damit ihr von Ihnen lernen könnt. Wenn ihr inhaltlich überhaupt nichts verstehen solltet, aber dafür die Fehlerkultur mitnehmt, sind wir happy.
  - A propos Fehlerkultur: Wenn ihr etwas nicht versteht - fragt nach. Das hier soll alles werden, bloß kein Frontalunterricht, in dem ihr Dinge schlucken, verdauen, und nachher wieder auskotzen sollt. Uns geht es darum, dass ihr Spaß habt, und vor allem: etwas lernt. Und gerade dem stehen so ein paar Dinge im Weg, wie: Schon beim ersten kleinen Fitzelchen Syntax zu scheitern. Also: Wenn ihr jetzt gleich bei euren Inspirern seid, und euer Programm nicht funktioniert, was macht ihr dann? Richtig. Fragt nach.

- grober inhaltlicher Überblick
- zeitlicher Überblick
- Einmal kurz die Technologie durchgehen

### Vorstellungsrunde 5 Min – Kleingruppe

- Stellt euch selbst vor. (2-3 Sätze)
  - Wie seid ihr zur IT gekommen?
  - Was habt ihr gemacht, als ihr so alt wart wie die Teilnehmenden?
- Lasst die Teilnehmenden sich vorstellen.
  - Wie heißt du?
  - Wieso bist du hier?
  - Hast du schonmal programmiert?
  - Was erhoffst du dir von diesem Kurs?
- Wichtig: Achtet darauf, dass die Kinder sich nur mit Vornamen vorstellen (Datenschutz :)

### Programmieren 3h + 30min Pause – Kleingruppe

#### Angeleiteter Teil: "Wir programmieren Pong"

In diesem Teil richten wir uns nach dem Pong-Tutorial auf der Seite <https://scratch.mit.edu/projects/editor>. Ihr findet es über „Tutorien > Pong-Spiel“. Dieser Teil ist relativ eng angeleitet. Teilt euren Bildschirm, programmiert vor und lasst den Teilnehmenden danach Zeit euch nachzuprogrammieren und auch eigene Ideen und Ansätze auszuprobieren.

Unser Programm besteht am Ende aus drei Gegenständen: Ball, Paddle und Linie. Im Weiteren werden diese Teile einzeln programmiert.

#### Vorstellen des Endresultats

Teilt euren Bildschirm und zeigt ihnen **dieses** Beispiel-Spiel, damit sich die Teilnehmenden gut vorstellen können, woran wir die nächste Zeit arbeiten werden. Lasst sie kurz überlegen, wie sie selbst dieses Spiel modifizieren können und was sie gerne Kreatives daran ändern möchten. Die Kinder sollen explizit dazu angeregt werden kreativ zu werden, was die Wahl von Hintergrund und Kostümen angeht.

#### Projekt erstellen

Jetzt bekommen die Teilnehmenden den Link für den Online Editor: <https://scratch.mit.edu/projects/editor/>  
Bei Scratch ist es für die Teilnehmenden vor allem wichtig viel selbst auszuprobieren, da sie nur so die Konzepte richtig begreifen. Es geht also weniger darum, den von uns vorgeschlagenen Code genauso nachher auf den Bildschirmen der Kids zu sehen, sondern darum, grundlegende Konzepte anhand dieses Codes zu erlernen. Fangt daher gerne damit an, gemeinsam mit der Webseite zu experimentieren und erklärt währenddessen, was ihr gerade macht. Gebt ihnen immer genug Zeit selbst Dinge auszuprobieren und mit ihrem Projekt herumzuspielen. Achtet darauf, dass alle die Seite geöffnet haben, bevor ihr weitermacht.



Data Club Lüneburg



### Eigene Kostüme, eigenen Hintergrund

Lasst die Teilnehmenden in ihrem eigenen Editor einen eigenen Hintergrund wählen. Dies geht unten rechts unter „Bühnenbild wählen“.

Nun sollen die Teilnehmenden ihren Ball anpassen. Dafür gehen sie unten rechts auf Figur wählen und wählen eine neue Figur aus. Die Katze, die standardmäßig auf der Bühne erscheint, kann gelöscht werden. Wählt einen Ball aus.

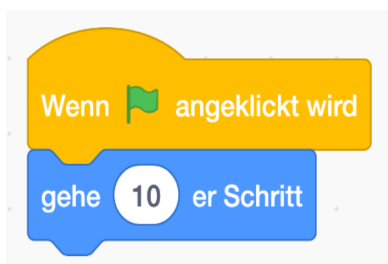
### Funktionen aufrufen

- Jeder Code in Scratch braucht einen Auslöser (Klick, grüne Flagge, Events)
- (Beispiel aus dem "echten" Leben: Youtube - Play Button; Ampel - Knopf drücken; intelligente Rollläden - Uhrzeit)
- Unser "Startereignis" in Scratch ist der Klick auf die grüne Flagge oben links über dem Bühnenbild.

Fügt den „Wenn grüne Flagge angeklickt wird-Block“ hinzu. Code, den wir nun an den diesen Block anheften, wird ausgeführt, sobald die grüne Flagge gedrückt wird.

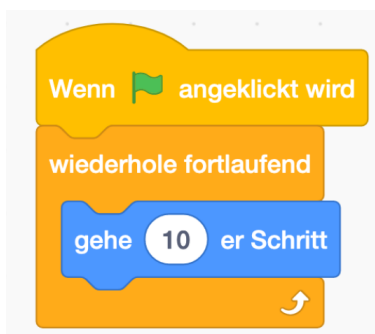
### Bewegung (Ball)

Fügt als Nächstes Bewegungen hinzu. Zeigt ihnen dafür den „gehe 10er Schritt-Block“. Lasst die Kinder ausprobieren, ob sie es hinbekommen, dass der Ball sich ein bisschen nach rechts bewegt.



### Schleifen (Ball)

Um nicht die ganze Zeit klicken zu müssen, können wir Schleifen verwenden. Da es (noch) keine Endbedingung für unser Spiel gibt, erstellen wir eine Endlosschleife um den Bewegungsblock.



Lasst die Teilnehmenden den Code ausführen. Der Ball schießt plötzlich nach rechts an den Rand und bewegt sich dann gar nicht mehr. Lasst sie kurz überlegen, wie wir das ändern könnten.

- Wie hätten wir gerne, dass der Ball sich verhält?
- Wann sollte der Ball seine Richtung ändern?
- In welche Richtung sollte der Ball sich dann bewegen?

Im Idealfall sollten sie darauf kommen, dass der Ball von der Wand abprallen sollte.

### If-Bedingungen (Ball)

Manche Programmteile wollen wir nur unter bestimmten Bedingungen ausführen.

- “Wenn es regnet, ziehe ich eine Regenjacke an. Sonst nicht.”
- In unserem Fall “Wenn der Ball die Wand berührt, dann ...”
- “... pralle vom Rand ab”

Fügt folgenden Code zu eurem Programm hinzu:



Hier ist ein guter Zeitpunkt, um auf die verschiedenen Kategorien (Farben) von Bausteinen einzugehen. Stellt vielleicht Fragen, wo die Teilnehmenden verschiedene Bausteine vermuten würden.

### Bewegung steuern über die Maus (Paddle)

Fügt ein Paddle hinzu (unten rechts: neue Figur hinzufügen). Der\*die Spieler\*in bewegt nachher das Paddle, damit der Ball nicht den Boden berührt. Um das Paddle zu programmieren, muss man dieses unten auswählen. Im Programmierfeld rechts oben sieht man das Bild des Gegenstandes, den man gerade programmiert.

Da wir das Paddle mit der Maus steuern wollen, setzen wir die x-Position des Paddles auf die x-Koordinate unserer Maus.

Verwendet dafür folgenden Code:

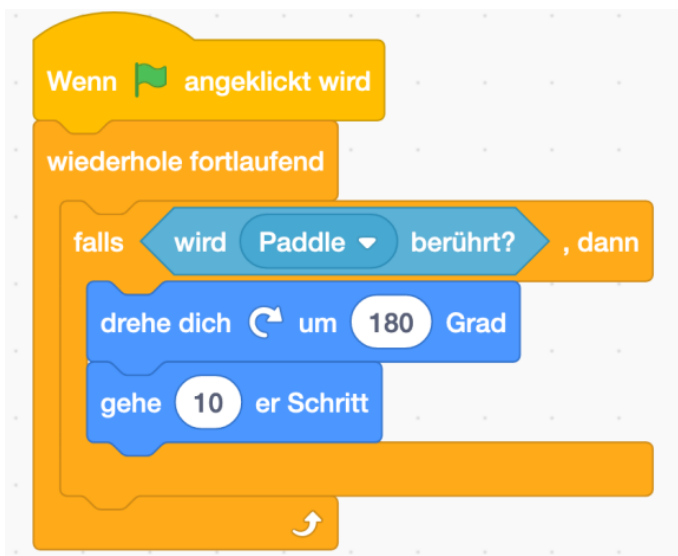







```
Wenn  angeklickt wird
wiederhole fortlaufend
  setze x auf 

```

### Interaktion zwischen Objekten (Ball)

Bisher fliegt der Ball durch das Paddle hindurch. Der Ball und das Paddle interagieren nicht miteinander. Fügt folgenden Code zu dem Ball hinzu und erklärt jeden Baustein:



```
Wenn  angeklickt wird
wiederhole fortlaufend
  falls  berührt? , dann
    drehe dich  um  Grad
    gehe  er Schritt

```

### Endbedingung (Linie)

Bisher fehlen dem Bewegungsablauf noch zwei Dinge:

- Das Spiel endet nicht, sobald der Ball den unteren Rand berührt

Dafür fügen wir am unteren Rand eine Linie hinzu (auch bei Figuren zu finden). Genau wie bei dem Paddle müssen wir jetzt für die Linie Code schreiben.



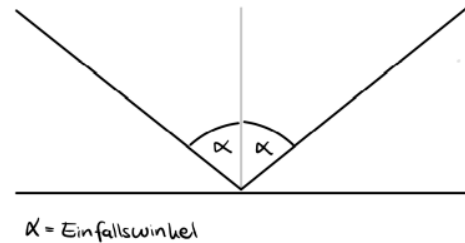
Ab hier funktioniert das Spiel grundsätzlich. Bevor die Kids komplett frei arbeiten, bietet es sich an noch zwei kleine Erweiterungen vorzunehmen.



### Korrektes Abprallen (Ball)

Hierfür muss man ein bisschen mit den Einfallswinkel- und Ausfallswinkeln herum überlegen.

- Der Ball soll die Richtung umkehren (drehe dich um  $180^\circ$ )
- Aber nicht genau dieselbe Richtung zurückfliegen - da wir hier eine "richtige" Physik wollen, wollen wir, dass Einfallswinkel = Ausfallswinkel gilt. Um das zu implementieren, machen wir uns die Variable Richtung (in der Skizze alpha) zunutze. ( $-2 * \text{Richtung}$ )
- Um ein bisschen Spannung reinzubringen, soll sich die Richtung um eine Zufallszahl variabel verändern.



### Spawnen (Ball)

Damit der Ball immer an einem festen Punkt startet und das Spiel fair ist, kann man den folgenden Block vor der Endlosschleife im Ball hinzufügen:



### Erweiterungen

Ab hier könnt ihr die Kids komplett frei arbeiten lassen, und sie hauptsächlich dabei unterstützen, ihre eigenen Ideen umzusetzen. Als Erstes können die Teilnehmenden noch einmal mit Kostümen, Hintergründen, Geschwindigkeit, Farben o.Ä. herumspielen. Das lockert die Atmosphäre auf und sorgt für schnelle Erfolgserlebnisse.

Einige weiterführende Ideen:

- zusätzliche Bälle
- einen Multiplayer-Modus (oben ein weiteres Line+Paddle-Paar, die Steuerung dann über Pfeil / WASD)
- Score/Highscore einführen (dazu braucht man Variablen, das Ganze ist im Scratch Tutorial gut umgesetzt)
- Ton, wenn der Ball das Paddle berührt (auch im Tutorial zu finden)

Außerdem gibt es bei Scratch noch einen ganzen Haufen weiterer Tutorials unter Tutorien (über den Online-Editor zu finden)

### **Interne Abschlussrunde 5 Min – Kleingruppe**

- Wenn ihr eurer Gruppe am Ende noch Feedback geben wollt, dann nehmt euch gerne jetzt Zeit dafür
- Ihr solltet eine Person bestimmen, die etwas von eurer Arbeit vorstellt.
- Es gibt aber nach dieser Abschlussrunde noch eine im großen Plenum.

### **Abschlussrunde 20 Min - Plenum - Haupt Inspirer**

- Die Schüler:innen stellen ihre Projekte vor (einer pro Gruppe), und erzählen ein bisschen, was sie so gemacht haben.
- Die Inspirer reden über sich, wie sie zur IT gekommen sind, was ihnen daran Spaß macht, wieso sie es weiterhin machen wollen, und was sie den Teilnehmenden gerne mitgeben möchten.
- Data Club-Orga & Feedbackbögen
- Der/Die Lehrer:in bekommt kurz das Wort.
- Der Hauptinspirer redet nochmal kurz darüber, welche Möglichkeiten es gibt sich noch mehr mit Programmieren auseinander zu setzen. Wir verteilen den Feedbackbogen-Link des Data Clubs Lüneburg an die Teilnehmer:innen und den/die Lehrer:in