

Scratch Tages-Workshop

Übersicht

Warum Scratch? Die grafische Programmiersprache Scratch funktioniert wie ein Lego-Baukasten und fördert auf spielerische Weise das logische, mathematische, analytische und algorithmische Denken. Indem man bunte Puzzleteile in der richtigen Reihenfolge anordnet, entstehen erste eigene Geschichte, Animationen und Spiele. Man erzielt schnell erste Erfolge und bekommt die Grundlagen des Programmierens intuitiv vermittelt.

Daher richtet sich dieser Tages-Workshop in erster Linie an Jugendliche ohne Programmiererfahrung im Alter von ca. 12 bis 16 Jahren.

Anmerkung: An unserem Workshop nahmen auch computeraffine 10-Jährige teil und hatten sehr viel Spaß.

Lizenz

CC BY 4.0 Jugend hackt Lab Heilbronn

Zielgruppe

12+

Kategorie

Coding

Online-Version mit Links zu weiteren zugehörigen Dokumenten

<https://jugendhackt.org/oer/projekte/scratch/>

Aufbau

Was brauche ich dafür?	2
Wie lange dauert der Workshop?	2
1. Ankommen, Begrüßen, Kennenlernen	3
1.1. Workshop-Start mit einem Energizer	3
1.2 Optional: Orientierungs- und Auflockerungsfragen	3
2. Scratch - Theorie und Vorbereitung des praktischen Teils	4
2.1 Was ist eigentlich Scratch?	4
2.2 Erstellung eines Scratch-Accounts	4
2.3 Einführung in die Entwicklungsumgebung	5
3. Scratch - Praktischer Teil	6
3.1 Namensprojekt	6
3.2 Der Ballon	8
3.3 Pong-Projekt	10
3.4. Abschluss (optional, wenn als Einzelworkshop konzipiert)	14
4. Jumping Game	14
Weitere Anmerkungen	20

Was brauche ich dafür?

- Hardware: Notebooks/PCs
- sonstige Geräte: eventuell Beamer und Leinwand bzw. interaktives Whiteboard
- Software: Scratch ist eine offene Online-Plattform. Es wird lediglich ein Account benötigt.
- Personenzahl: empfehlenswert maximal 15 TN und drei Mentor*innen

Wie lange dauert der Workshop?

Der Workshop kann sowohl remote als auch in Präsenz stattfinden.

In Präsenz ist es denkbar, die Inhalte entweder als Tagesworkshop von rund sechs Stunden oder als Wochenendformat verteilt auf zwei Tage anzubieten. Da man remote mehr Zeit für die technische Vorbereitung sowie Pausen und Aktivierungsspiele einplanen sollte, um die virtuelle Distanz zu überwinden, raten wir hier zu zwei einzelnen Workshops. Ein detaillierter Ablaufplan findet sich im verlinkten ZIM-Papier, das die **Ziele**, **Inhalte** und **Methoden** genauer beschreibt. Pausen und Auflockerungen (bzw. Energizer) sind explizit mit im Zeitplan aufgenommen, um den Workshop so nachhaltig und anregend wie möglich zu gestalten.

1. Ankommen, Begrüßen, Kennenlernen

Die Begrüßung und Einführung in die Workshop-Formate folgt bei Jugend hackt einem festen Standard, daher wird dieser nicht im Detail vorgestellt. Workshop-Leitende sollten jedoch ein paar Minuten einplanen, um sich selbst bzw. die Organisation, für die das Format stattfindet, vorzustellen. Wichtig ist es zudem, den Jugendlichen zu vermitteln, was die Ziele und der Mehrwert des Workshops sind.

1.1. Workshop-Start mit einem Energizer

Energizer sind einfache Aktivierungsspiele. Aktivierungsspiele sind nicht der eigentliche Gegenstand des Unterrichts, sondern ein unterstützendes pädagogisches Mittel, das zu unterschiedlichen Zwecken – Kennenlernen, Gruppen-/Teambildung, Entspannung, Konzentration, Koordinations- und Bewegungsförderung – eingesetzt werden kann.

Wir empfehlen vor allem im digitalen Kontext einen Energizer, den wir “Kamera an!” nennen, da wir die Erfahrung gemacht haben, dass die Jugendlichen sehr zurückhaltend dabei sind, die eigene Kamera während des Workshops zu nutzen. Der Workshop-Speaker denkt sich im Vorhinein einige Fragen aus und stellt sie. Alle, die die Frage mit “ja” beantworten können, müssen für einige Sekunden ihre Kamera anmachen. Für die nächste Frage wird die Kamera wieder ausgeschaltet.

Ein Beispiel: Der*Die Workshop-Speaker fragt: *“Wer geht gerne Eislaufen?”*
Eine Teilnehmerin hat Lust auf Eislaufen? Ja? Dann Kamera an!

Dieses Spiel lässt sich auch in Präsenz spielen, nur dass hier alle, die eine Frage bejahen können, einen Schritt vortreten.

Als Alternative, die sowohl digital als auch in Präsenz funktioniert, können Zeichnungen erraten werden. In Präsenz kann man hierfür ein Whiteboard nutzen. Online bieten Dienste wie Big Blue Button eine Malfunktion an, so dass alle Teilnehmenden auf die aktuelle Folie malen können. Auch hier bereitet man Fragen vor:

Ein Beispiel: Der*Die Workshop-Leitende fragt: Was hast du zu Weihnachten bekommen?
Der*Die Jugendliche malt etwas, alle Anderen müssen es erraten.

1.2 Optional: Orientierungs- und Auflockerungsfragen

Diese Fragen dienen dazu, den Wissensstand aller Teilnehmer*innen abzufragen. Gleichzeitig kann man vermitteln, dass fehlende Vorerfahrungen bei Scratch kein Problem darstellen. Folgende Fragen kann man deshalb stellen:

- *Hat jemand hier Erfahrung mit Scratch?*
- *Wenn ja, welche? Hast du schonmal etwas erstellt, was du den anderen zeigen willst?*
- *Wer hat schon programmiert?*

- Was wisst ihr über das Programmieren?
- Was stellt ihr euch vor, gehört zum Programmieren dazu?
- etc.

2. Scratch - Theorie und Vorbereitung des praktischen Teils

2.1 Was ist eigentlich Scratch?

Das folgende YouTube-Video bietet sich für einen verständlichen Einblick in die Plattform an:

<https://www.youtube.com/watch?v=SUSSsZLxjDs&t=182s>

(Alternativ kann der*die Workshop-Leitende selbst einige einführende Worte sagen)

Im Anschluss geht es für alle zur Plattform selbst unter dem Link: <https://scratch.mit.edu/>.

Folgendes sollte oben auf der offenen Website zu sehen sein:



Zuerst stellt man die einzelnen Bereiche vor: Unter dem Button *“Entdecken”* findet man Projekte von anderen Scratch-Usern. Nehmt euch gerne die Zeit und probiert das ein oder andere aus. Hier kann man anhand der Zielgruppe entscheiden, wie viel Zeit man dafür einplant

Unter dem Button *“Ideen”* findet man viele Anleitungen und hilfreiche Tutorials zum Nachbauen.

Wenn ihr wirklich loslegen und eigene Programm schreiben wollt, dann klickt auf den Button *“Entwickeln”*.

2.2 Erstellung eines Scratch-Accounts

Um Scratch nutzen zu können, braucht es einen Scratch-Account. Dafür benötigt man eine gültige E-Mail Adresse. Ein*e Teilnehmende*r hat keine? Halte für diese Möglichkeit eigens angelegte Mailadressen bereit.

1. Klicke oben in der Leiste auf den Button *“Scratcher werden”*

A close-up image of the two buttons from the navigation bar: 'Scratcher werden' and 'Anmelden', both in white text on a blue background.

2. Wähle einen Benutzernamen und ein Passwort.

Scratcher werden
Erstelle Projekte, teile Ideen, finde Freunde. Es ist kostenlos!

Wähle einen Benutzernamen

Benutze nicht deinen echten Namen

Passwort erstellen

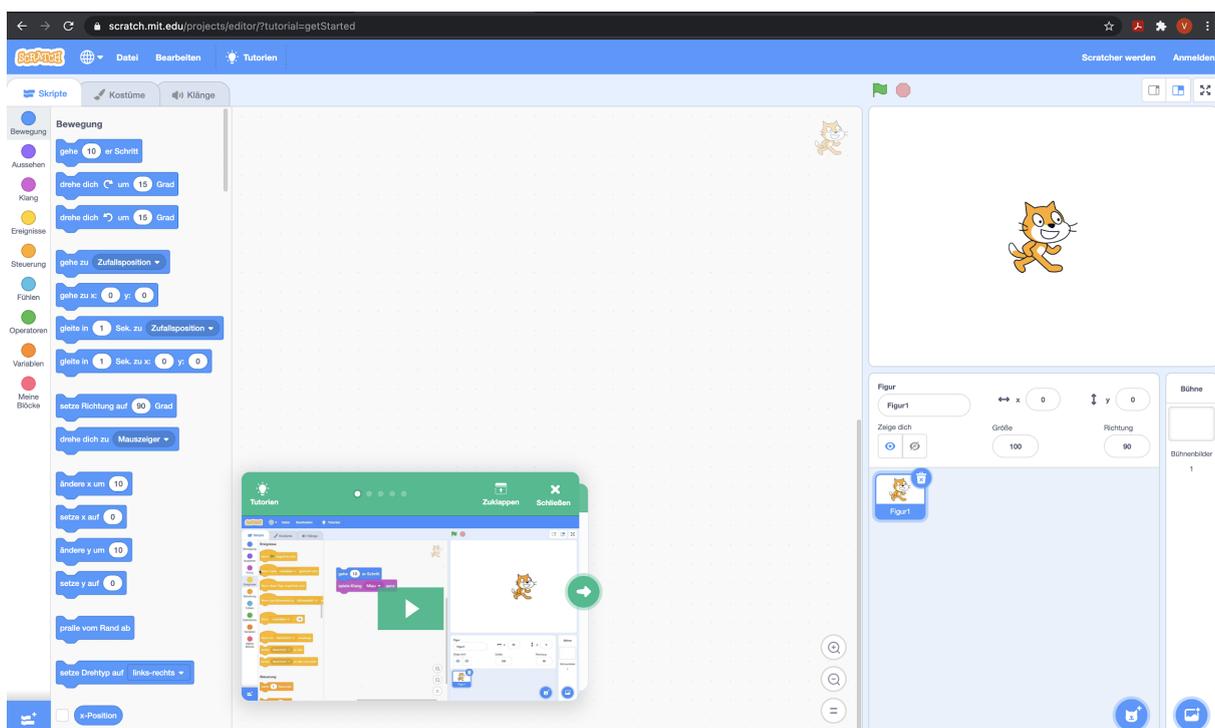
Passwort anzeigen

Weiter

3. Man erhält im E-Mail-Programm eine Mail. Nach der Bestätigung kann es losgehen.

2.3 Einführung in die Entwicklungsumgebung

Nun geht es darum, die Umgebung, in der alle Teilnehmenden ihre eigenen Animationen und Spiele entwickeln können, kennenzulernen. Dazu geht es auf den Button "Entwickeln". Folgendes solltet ihr sehen:



Ganz links erkennt ihr die verschiedenen Puzzleteile, aus denen man Stück für Stück eigene Projekte “zusammenstecken” kann. Schaut genau hin: es gibt mehrere Farben, die verschiedene Kategorien kennzeichnen, zum Beispiel “Aussehen”, “Bewegung”, “Töne” etc.

Für Workshop-Leitende, die zuvor nicht selbst mit Scratch gearbeitet haben, empfehlen sich die folgenden zwei Videos, dessen Informationen an die Teilnehmenden weitergegeben werden sollten:

<https://scratch.mit.edu/projects/editor/?tutorial=getStarted>

<https://www.youtube.com/watch?v=jYbTamyEXxI> .

Viele Funktionen werden in den Projekten, die im praktischen Teil folgen, angewendet und erklären sich spätestens dann. Grundlegend sollte jedoch gezeigt werden, wie man die einzelnen Blöcke verschiebt, um sie auszuprobieren, wie man das komplette Projekt startet, um den aktuellen Stand zu prüfen und wie man diesen abspeichern kann.

(Bei Bedarf kann nun eine erste Pause gemacht werden)

3. Scratch - Praktischer Teil

3.1 Namensprojekt

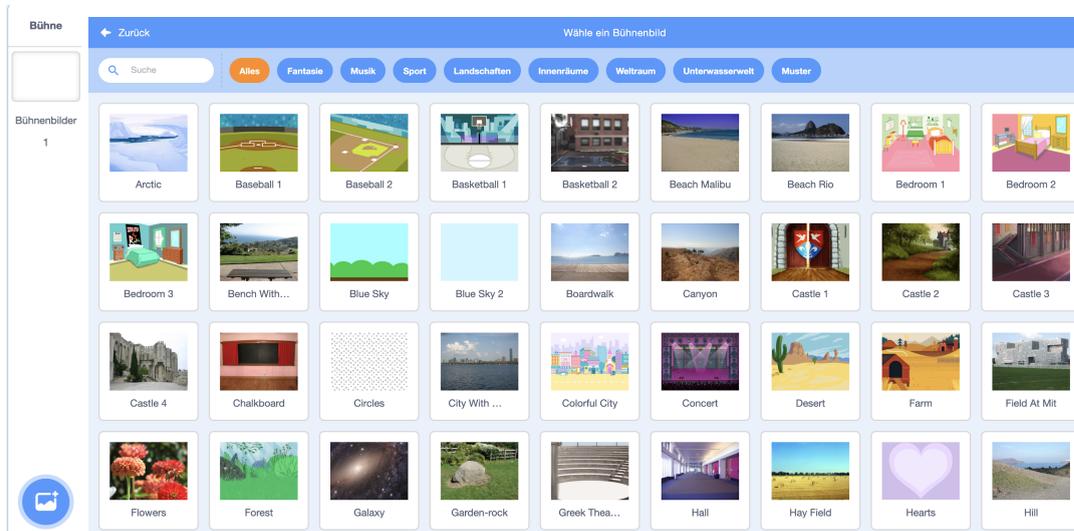
Als einfachste Anwendung sollen die Teilnehmer*innen nun eine individualisierte Animation bauen. Sie schreiben ihren Namen und geben den einzelnen Buchstaben witzige Effekte. So lernen sie die ersten Funktionen wie “Klänge”, “Bewegung” kennen.

Im Folgenden ist ein Beispiel Schritt für Schritt erklärt. Dieses muss nicht eins zu eins nachgebaut werden. Fühlt euch frei und lasst der Kreativität ihren Lauf, denn Scratch funktioniert vor allem über das Ausprobieren.

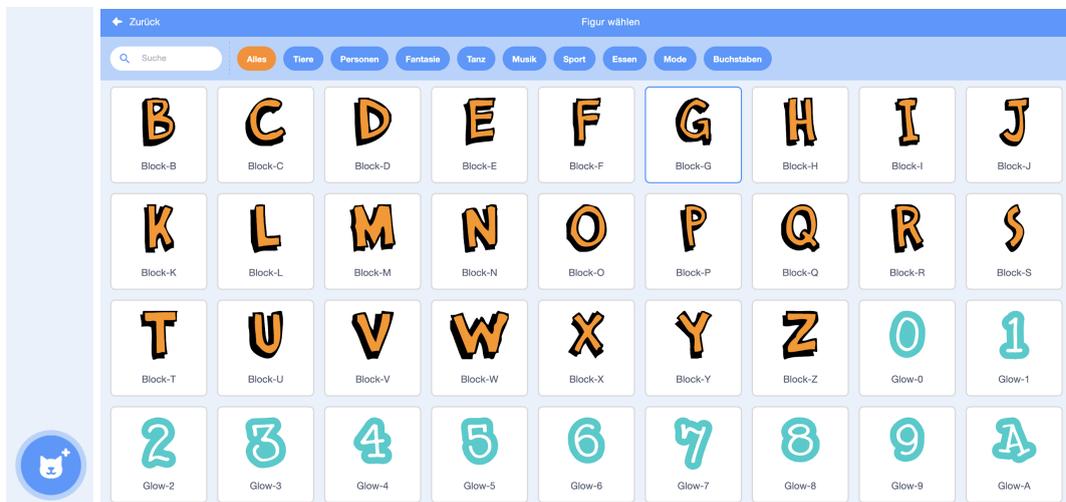
1. Löscht die Katze durch Klicken auf das Mülltonnensymbol



- Wählt ein Hintergrund aus. Klickt dazu auf das Bühnenbildsymbol (rechts unten) und auf den Hintergrund deiner Wahl. Schnelle Scratcher können auch einen eigenen Hintergrund hochladen.



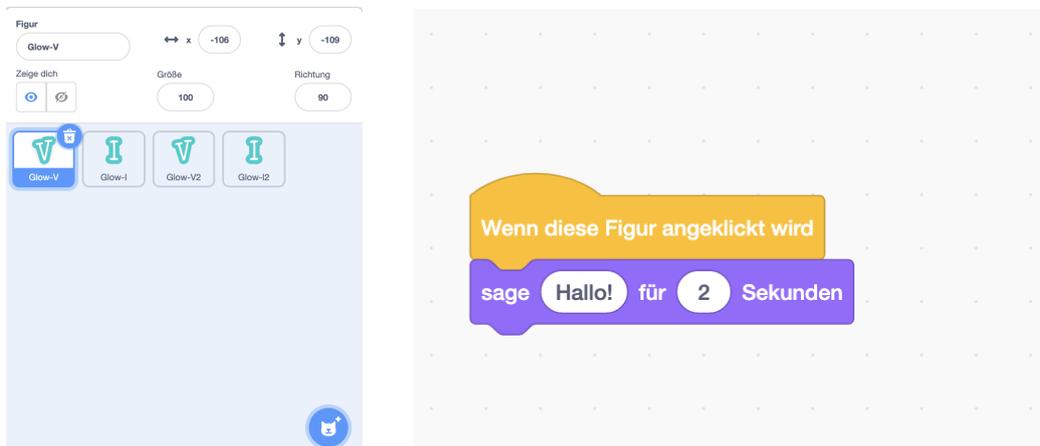
- Jetzt müsst ihr euch alle Buchstaben für den jeweiligen Namen zusammensuchen. Klickt dafür auf den Button "Figur wählen" und sucht eure Buchstaben. Diesen Prozess müsst ihr für jeden Buchstaben wiederholen.



- Noch sind alle Buchstaben verstreut in deinem Anzeige-Fenster. Klickt einfach auf den jeweiligen Buchstaben und schiebe ihn an die richtige Stelle.



- Jetzt wollen wir, dass unsere Buchstaben auch etwas tun können. Das müssen wir auch wieder für jeden Buchstaben einzeln machen. Klicke dafür im Figurenfenster den ersten Buchstaben an. Wir wollen, dass dieser Buchstabe kurz "Hallo" sagt, wenn er angeklickt wird. Dafür benutzt ihr einen gelben Ereignisstein, auf dem steht "Wenn diese Figur angeklickt wird" und einen lilanen Klangstein "Sage Hallo für 2 Sekunden". Testet es aus. Sagt der Buchstabe "Hallo", wenn er angeklickt wird?



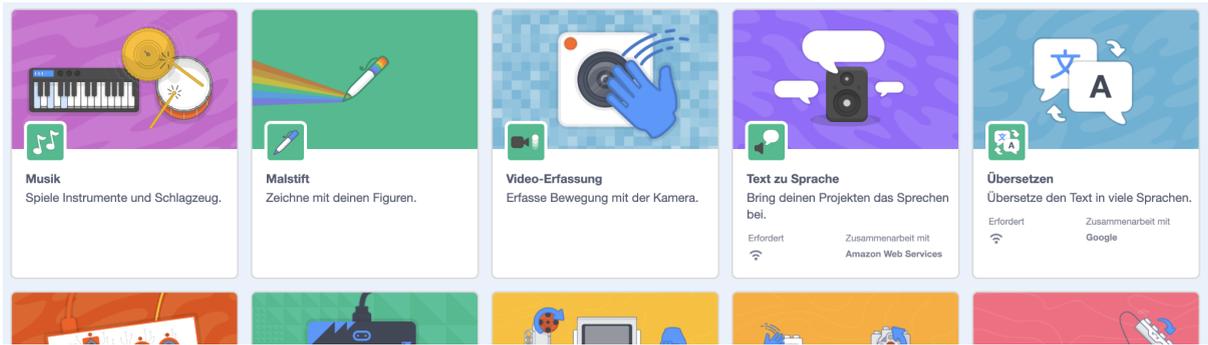
- Testet alle möglichen Dinge aus und baut für alle eure Buchstaben witzige Animationen. Ein Beispiel findest du hier: <https://scratch.mit.edu/projects/493484778/> Alle Teilnehmer*innen sollten sich die Zeit nehmen, um an ihrem eigenen Namensprojekt herumzuprobieren. Referent*innen und Mentor*innen stehen nur unterstützend zur Seite.
- Zeigt, was ihr gemacht habt. In Präsenz können alle von Arbeitsplatz zu Arbeitsplatz gehen. Remote kann der Bildschirm freigegeben werden, um das eigene Projekt zu zeigen.

3.2 Der Ballon

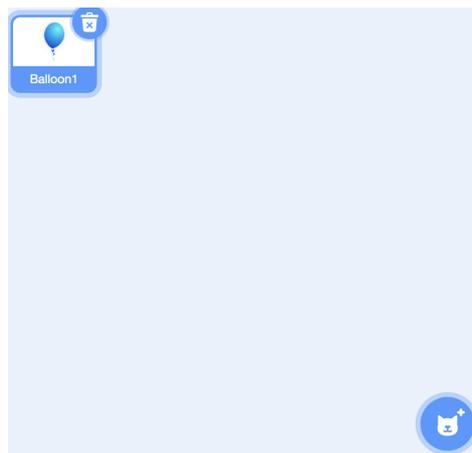
- Wichtig ist vor allem, dass ihr die Videoerfassungserweiterung einstellt. Klicke dazu erst auf den diesen Knopf unter den bunten Blöcken:



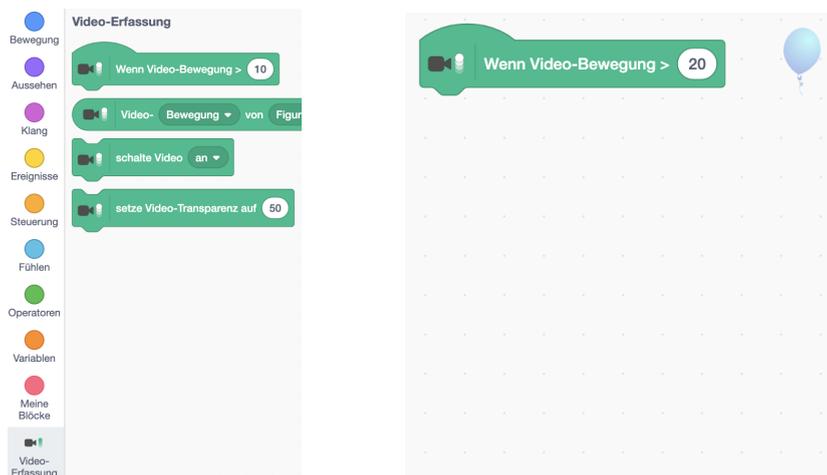
Wähle jetzt die Erweiterung Videoerfassung aus (siehe Mitte).



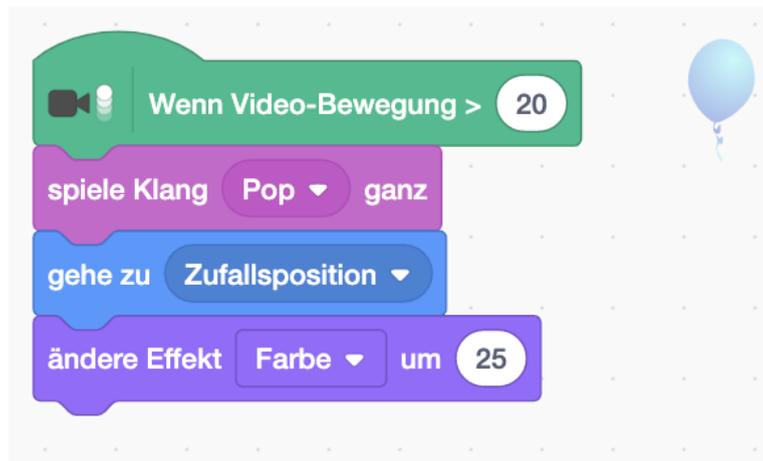
2. Füge jetzt einen Ballon ins Bild ein. Gehe dabei genauso vor, wie oben bei dem Namensprojekt. Lösche die Default-Katze und suche unten rechts unter “Figur wählen” nach dem Ballon (Tipp: wenn du ihn nicht findest gib einfach “Ballon” in der Suchleiste ein). Dein Figurenfeld sollte jetzt so aussehen.



3. Jetzt wird das erste kleine Spiel geschrieben. Sucht unter dem Punkt “Videoerfassung” den Block “Wenn Video Bewegung >”, ziehe ihn in das Bearbeiten-Feld und schreibe statt 10, 20 hinein.



- Der Rest funktioniert so wie ihr es bereits kennt. Sucht unter den lilafarbenen Klangbausteinen den Stein “*Spiele Klang Pop ganz*” und füge ihn unter dem Videostein ein. Dann suchst du unter den blauen Bewegungsbausteinen den Stein “*gehe zu Zufallsposition*” und füge ihn unter dem lilafarbenen Stein ein. Als letztes brauchst du einen (ebenfalls in einem Lila-Ton gehaltenen) Aussehen-Block. Finde “*ändere Effekt Farbe um 25*” und füge ihn am Ende ein. Dein Programm sollte jetzt so aussehen:



- Jetzt könnt ihr den Ballon platzen lassen! Berührt ihn mit eurem Finger und PUFF! Dieses kleine Spiel funktioniert in Fortgeschrittenen-Workshops zu Scratch übrigens auch super als Energizer.

3.3 Pong-Projekt

Beinahe jede*r kennt den Spieleklassiker Pong. Das Spielprinzip ist sehr einfach. Hier zum Schmunzeln das Video des Originals: <https://www.youtube.com/watch?v=fiShX2pTz9A>

Ihr sollt das Ganze jetzt im Einzelplayer-Modul umsetzen. Ziel ist: Ball nicht fallen lassen!

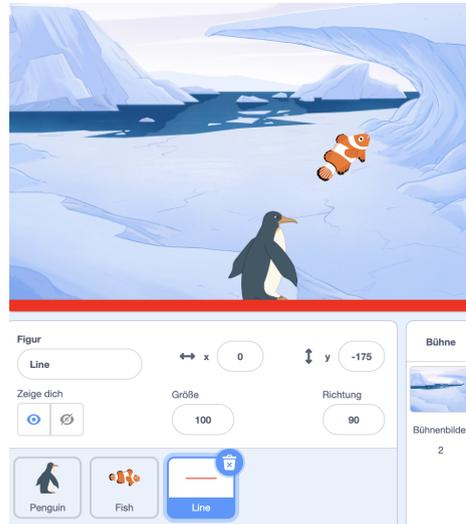
An diesem Beispielprojekt, das eine unserer Mentorinnen gebaut hat, zeigen wir dir, wie es geht:

<https://scratch.mit.edu/projects/493513953/>

- Als erstes müsst ihr euch ein Spielkonzept überlegen. Möchtet ihr einen Drachen, der einen vergifteten Apfel in einer Burg jagt, dann sucht diese Figuren aus. Unsere Mentorin hat sich gedacht, dass sie mit einem Pinguin spielen möchte, der versucht, einen Fisch in einer Eislandschaft zu fangen. Also auf geht's: Erstellt ein neues Projekt, löscht die Default-Katze und sucht den passenden Hintergrund und die passenden Figuren nach dem altbekannten Konzept zusammen!



- Jetzt fragt ihr euch vielleicht: wann habe ich verloren? Ganz einfach: wenn das gejagte Objekt die rote Linie trifft. Welche rote Linie? Die müsst ihr natürlich erstellen. Sucht unter “Figuren” die rote Linie aus, zieht sie groß und ans untere Ende des Spiele-Bildschirms. Das Jäger-Objekt (hier Pinguin) kurz darüber.



- Nun gehts ans Programm schreiben. Wir beginnen mit dem Jäger-Objekt. Im Beispiel ist das der Pinguin. Klickt also auf das Jäger-Objekt und zieht zuerst den hellorangenen Ereignisstein “wenn grüne Flagge angeklickt wird” ins Feld. So wird das Spiel immer gestartet, wenn auf die grüne Flagge rechts im Spielfenster geklickt wird. Dann sucht ihr unter den dunkelorangeen “Steuerungsbausteinen” den “wiederhole fortlaufend” Stein. Jetzt muss noch eingefügt werden, was der Pinguin fortlaufend machen soll. Ganz einfach: Er soll am Boden immer eurer Maus-Bewegung folgen. Sucht also unter Bewegung “setze x auf” und zieht ihn in den orangenen Stein. Also worauf müsst ihr x immer setzen? Genau: auf die “Maus x-Position”. Die findest du unter den blauen “Fühlen”-Bausteinen.

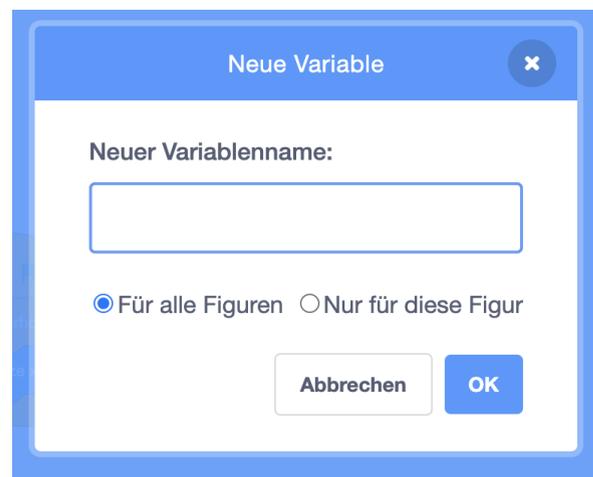


- Als nächstes widmet ihr euch dem gejagten Objekt. In diesem Fall dem Fisch. Das ist das Schwierigste, weshalb wir es in drei Schritte unterteilen.

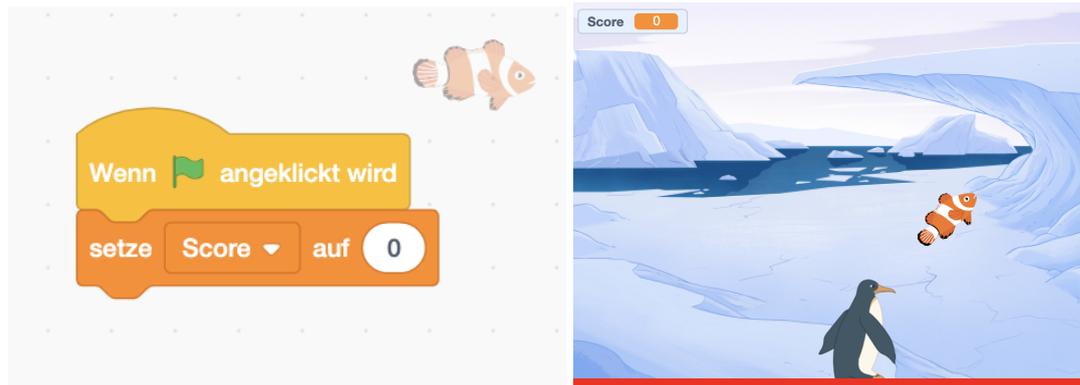
Schritt 1: Der Fisch soll sich die ganze Zeit bewegen und an den Ecken des Spielfeldrandes abprallen. Dazu sucht wieder den “wenn grüne Flagge angeklickt wird” Stein und zieht ihn in das Feld. Steckt darunter den blauen “setze Richtung auf 45 Grad” Block. Darunter kommt wieder der “wiederhole fortlaufend” Stein. In den muss man nur noch zwei blaue Bewegungs-Elemente packen. Als erstes “gehe 15er Schritte”, dann “pralle vom Rand ab”. Vergesst nicht, wenn die Werte mal nicht passen sollte, könnt ihr beinahe immer in das Feld klicken und sie ändern.



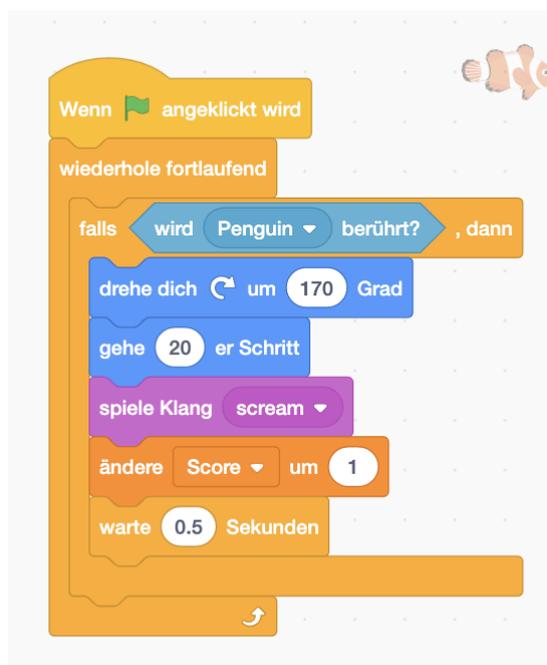
Schritt 2: Ihr wollt bestimmt wissen, wie oft ihr den Fisch daran hindern konntet, die rote Linie zu berühren!? Dafür erstellen wir eine neue Variable und nenne sie “Score”. Klicke auf den orangenen Button Variablen. Hier klickst du neue Variable und gibst ihr einen Namen.



Der Score soll am Anfang 0 sein. Dazu brauchst du (wer hätte es gedacht) den Grüne-Flagge-Stein. Zieht ihn in das Feld und setzt darunter den Variablen-Stein “Setze *myvariable* auf 0”. Wähle dann unter *myvariable* den Namen deiner Variable. Ihr solltet nun übrigens ein Feld mit *Score* im Spielfeld sehen. Zieht das einfach in die linke obere Ecke.



Schritt 3: Jetzt soll der Fisch aber noch vom Pinguin abprallen und das geht so: Wählt wieder den *“wenn grüne Flagge angeklickt wird”* Stein aus. Darunter kommt der bekannte *“wiederhole fortlaufend Stein”* und hinein steckt ihr den orangenen *“falls...dann”*. Jetzt könnt ihr noch das *“...”* mit dem hellblauen *“wird Mauszeiger berührt?”* Stein ersetzen und hier im Dropdown-Menü das Jäger-Objekt auswählen. Was soll der Fisch tun, wenn er den Pinguin berührt? Zuerst soll der Fisch sich um 170 Grad drehen, dann 20 Schritte weiter gehen (diese Zahlen beruhen auf eigenem Ausprobieren der idealen Werte). Außerdem soll der Score natürlich um eins nach oben gezählt werden, um den Punktestand richtig wiederzugeben. Damit das Ganze zeitlich passt, fehlt bei Steuerung ein *“warte 0.5 Sekunden”*. Das Ganze sollte dann so aussehen:



- Der letzte zu programmierende Schritt betrifft die rote Linie. Berührt der Fisch die Linie, dann ist das Spiel natürlich vorbei. Klickt im Figurenfeld auf die Linie und sucht zuerst unter Ereignisse *“wenn grüne Flagge angeklickt wird”*. Das ist der Beginner-Stein. Jetzt braucht ihr drei orangene Steuerungssteine. Zieht zuerst *“wiederhole fortlaufend”* unter

den gelben Stein, sucht dann “falls ... dann” und steckt ihn in den ersten orangenen Stein. Als letztes sucht ihr das “stoppe alles” Element und zieht es ganz in die Mitte. Fast fertig! Vielleicht habt ihr gesehen, dass bei “falls ... dann” ein weiteres Feld innen ist, dass ihr verändern könnt. Sucht dazu unter “Fühlen” den graublauen Stein mit “wird Mauszeiger berührt” und wählt im Dropdown-Menü den Namen vom gejagten Objekt aus. Bei uns ist das der Fisch. Keine Sorge, das ist viel leichter als es sich anhört, schaut euch zur Hilfe einfach das Bild an.



- Der letzte Schritt ist der beste: Einfach spielen! Wie lange schaffst du es?

3.4. Abschluss (optional, wenn als Einzelworkshop konzipiert)

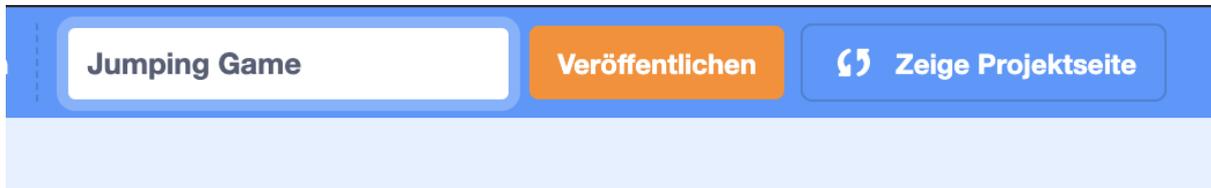
(ansonsten Pause)

Zeit für Feedback sollte immer sein. Hat es den Teilnehmenden Spaß gemacht? Haben sie alles verstanden, sind gut mitgekommen und haben etwas gelernt? Es gibt verschiedene Möglichkeiten, um das Feedback zu verschriftlichen, eine davon ist ein Feedback-Radar. Dieses kann entweder remote als Folie eingeblendet werden oder in Präsenz auf ein Whiteboard aufgemalt werden. Alle Teilnehmenden können dann ihr persönliches Feedback anonymisiert (mit jeweils eigener Farbe) dalassen.

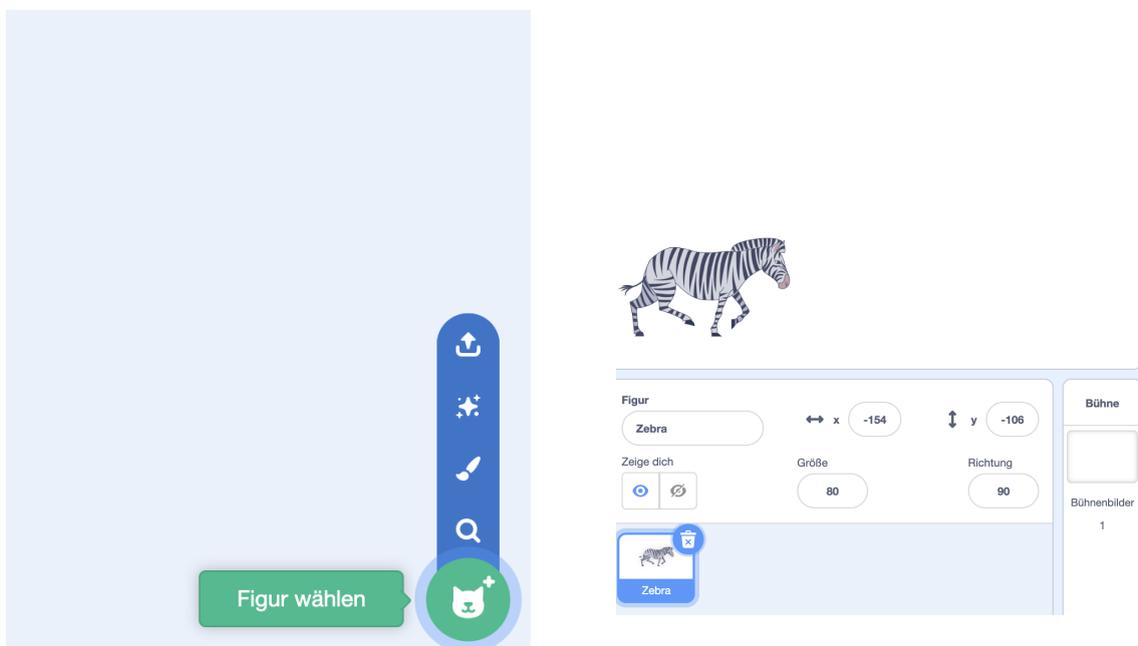
4. Jumping Game

Ihr lernt nun, wie man ein Jumping Game baut. Kennt ihr das lustige Dinosaurier-Spiel, das man bei Google spielen kann, wenn man wieder einmal kein Internet hat? Ja? Sowas wollen wir jetzt auch machen. Referent*innen können sich im Vorhinein das folgende Video ansehen, um die Logik des Jumping Game zu verstehen: <https://www.youtube.com/watch?v=1jHvXakt1qw>

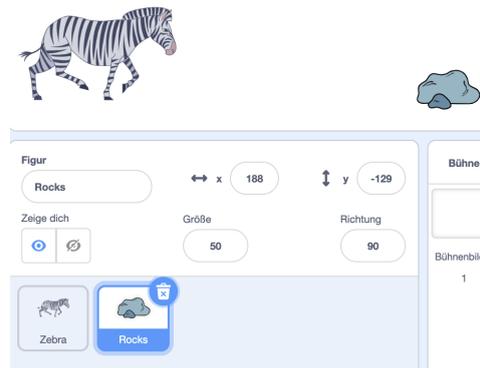
1. Erstellt ein neues Projekt und gebt ihm einen Titel.



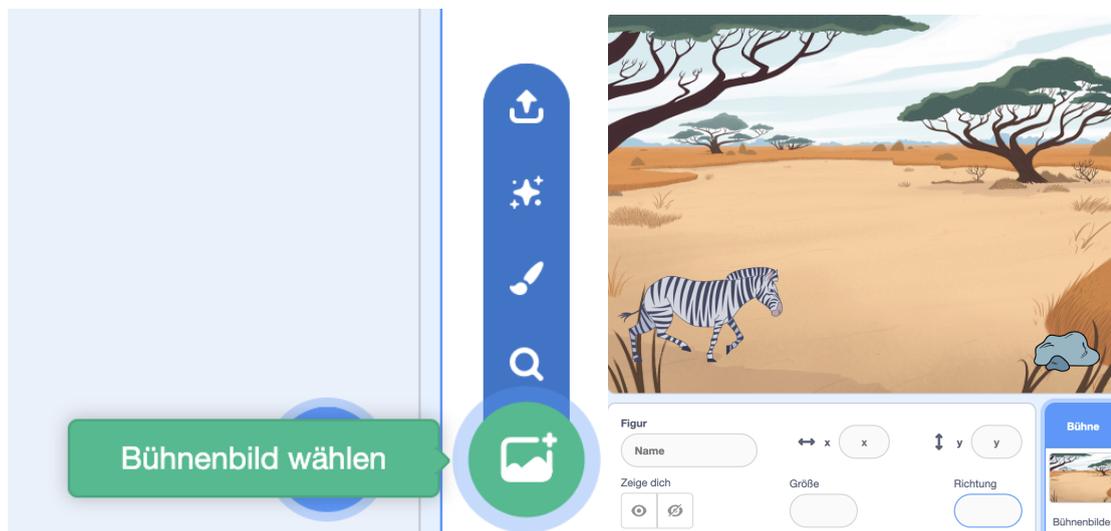
2. Löscht die Default-Katze und wählt ein Objekt, das springen soll. In unserem Beispiel ist es das Zebra. Zieht es in die linke untere Ecke des Spielfeldes und passt auf, dass es nicht zu groß ist. Die Größe kann man unter *Figur* > *Größe* ändern. Wir haben unser Zebra auf 80 statt 100 gesetzt.



3. Wählt jetzt ein Objekt, über das gesprungen werden soll. Genauso wie das Springer-Objekt findet ihr es unter *Figur wählen*. Wir wählen einen Stein und passen auch hier die Größe an, sodass unser Zebra gut darüber springen kann. Diesen Stein zieht man in die rechte untere Ecke des Spielfeldes.

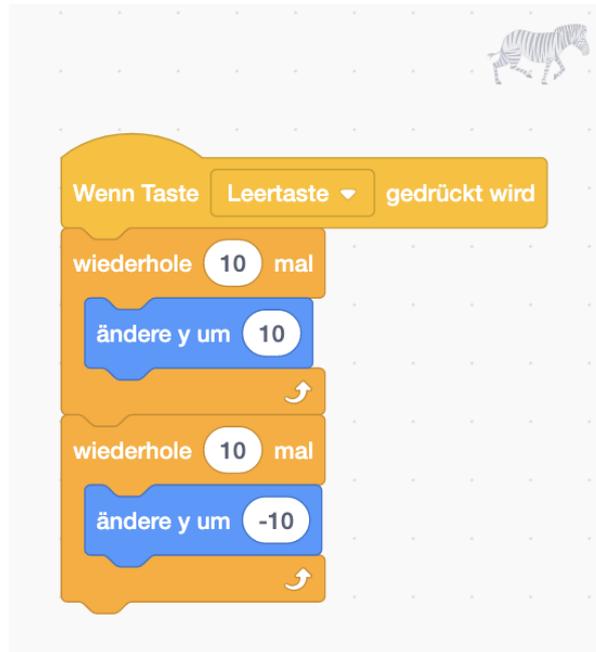


- Das sieht doch schon ganz gut aus, was noch fehlt ist ein Bühnenbild. Klickt dazu auf *Bühnenbild wählen* und sucht eines aus, das euch gefällt. Passend zum Zebra haben wir die Savanne gewählt.

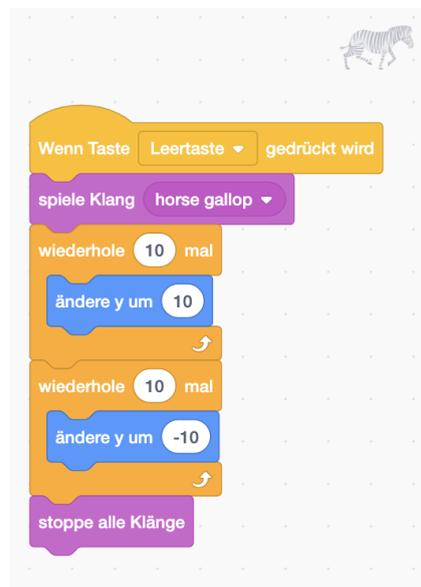


Jetzt gehen wir ans Coden. Das ganze Programm lässt sich in 5 Schritte unterteilen. Achtet darauf, dass ihr euch immer in der richtigen Figur befindet. Ihr seht in den folgenden Bildern in der oberen rechten Ecke immer, in welcher Figur sich der Code befindet.

- Bringt den Charakter zum Hüpfen, wenn ihr die Leertaste drückt. Er muss sich also von seinem Ausgangspunkt zu einem höheren Punkt bewegen: eben hüpfen. Bei Scratch wird für die Positionsbestimmung mit Koordinaten gearbeitet. Wählt den hellorange Steuerungsbaukasten "wenn Taste " " gedrückt wird" und puzzelt die restlichen Bedingungen dazu. Probiert gerne mit den Werten herum, damit ihr versteht, wie eure Figur reagiert und sich mit unterschiedlichen Vorgaben im Feld bewegt.



Das Ganze kann man dann übrigens noch spaßiger gestalten, indem man einen “Hüpf-Sound” einfügt:



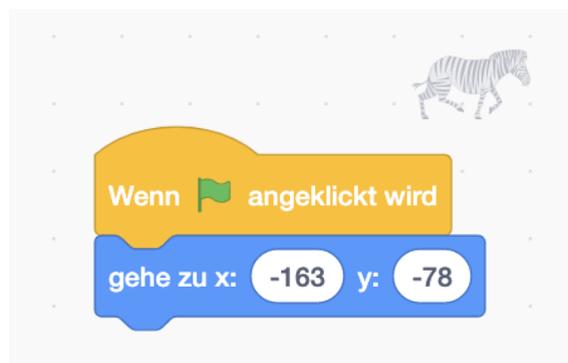
- Ergänzt sich bewegende Hindernisse. Das Hindernis soll ja regelmäßig von rechts nach links sliden anstatt nur einmal aufzutauchen und dann nie wieder. Tipp: Wenn du das Objekt an die Stelle legst, bevor du die blauen Steine einbaust, dann stellt dir Scratch automatisch die richtige Position ein.



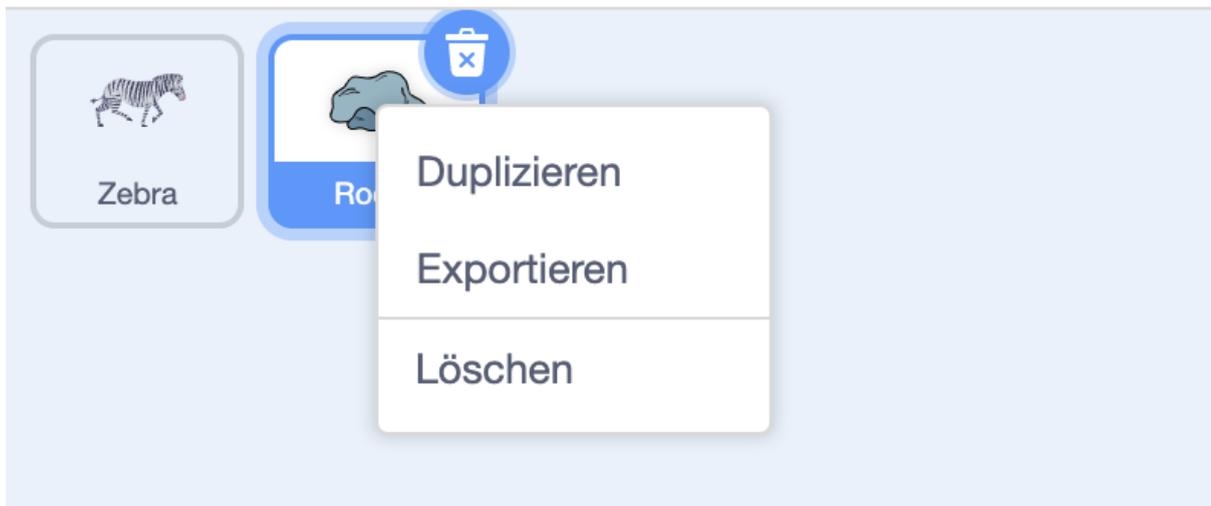
7. Beendet das Spiel. Natürlich nicht immer. Das Spiel soll enden, wenn die springende Figur das Hindernis berührt.



8. Jetzt müssen wir noch verhindern, dass die Figur mitten in der Luft stoppt, wenn das Spiel gestoppt wird. Dazu benötigen wir folgende Blöcke (Achtung: die sind im Entwickeln-Feld eurer Springerfigur):



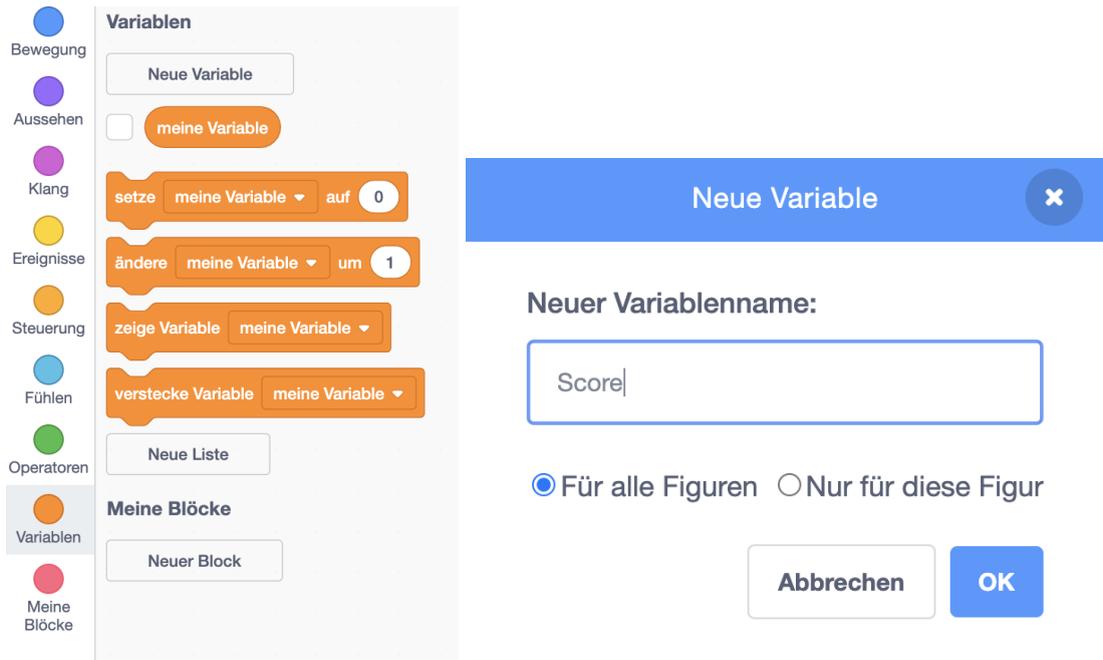
9. Baue mehr Hindernisse ein. Mit einem Hindernis, das sich immer gleich bewegt, wird das Ganze ja schnell langweilig. Deshalb brauchen wir mehr Hindernisse.



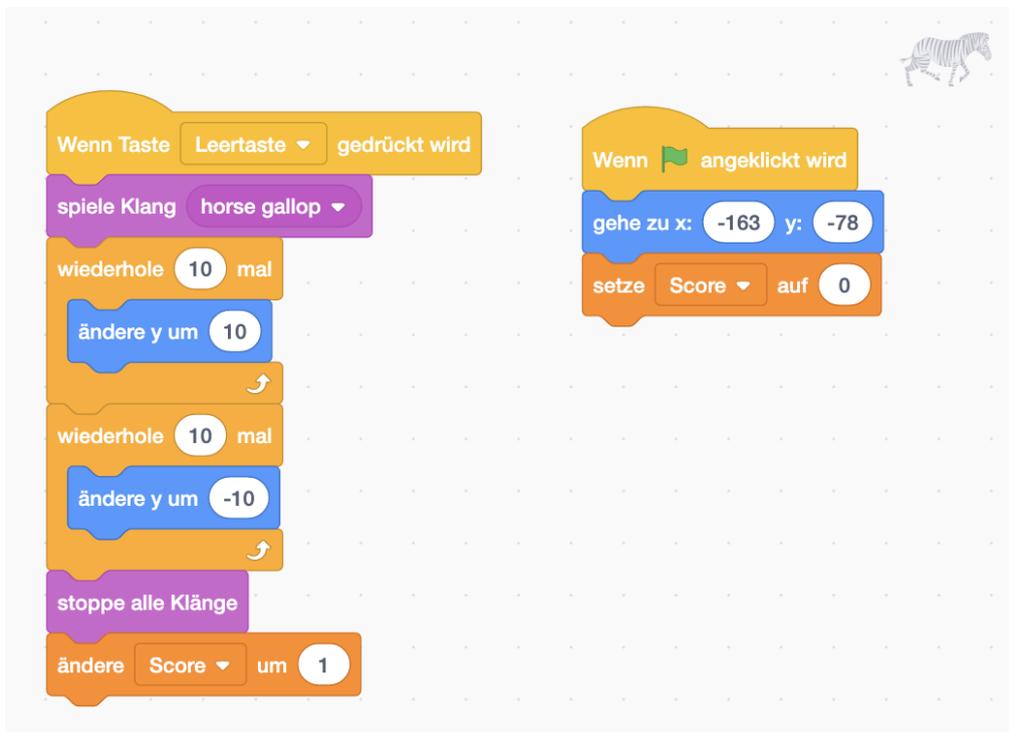
Einfach Rechtsklick auf das Hindernis und dann Duplizieren drücken. Im zweiten Stein müsst ihr nur drei Blöcke in den bestehenden Code einfügen, wodurch der zweite Stein immer verzögert zum ersten Stein kommt und die Verzögerung zudem variiert.



10. Baut einen Highscore ein. Die Figur soll immer einen Punkt erhalten, wenn sie über ein Objekt gesprungen ist. Dafür erstellt ihr zuerst eine "Neue Variable" und nennt sie Score:



Jetzt fehlt nur noch das Hochzählen! Dazu geht auf das Figuren-Entwickeln-Feld und ergänzt folgende Blöcke:



Vielleicht erinnert ihr euch: auch beim Pong-Spiel musstet ihr einen Score erstellen. Vielleicht bekommt der eine oder die andere das sogar ganz ohne Anleitung hin.

Fertig! Glückwunsch, ihr habt ein Jumping Game programmiert. Stellt es doch gegenseitig vor!

Weitere Anmerkungen

In unserem Workshop haben wir aufgrund der eher jüngeren Teilnehmenden sowie des Remote-Formats Schluss gemacht. Für einen Tagesworkshop bietet sich an, eine weitere Stunde anzuhängen und die Teilnehmenden nun ganz eigene Ideen entwickeln zu lassen. Dabei sollten Referent*innen und Mentor*innen unterstützend zur Seite stehen, um die Ideen möglichst schnell und problemlos umsetzen zu können.

Grundsätzlich sollten die Referent*innen dazu ermutigen, so viel wie möglich selbstständig zu arbeiten. Gerade nach den ersten zwei Anwendungen können Teilnehmende kleine Schritte bereits ohne große Hilfe umsetzen.

Anmerkung zu Remote-Angeboten:

Prinzipiell kann der Workshop beliebig gekürzt werden, zum Beispiel nach dem Namens- oder Ballonprojekt beendet werden. Bei besonders vielen Teilnehmer*innen bietet es sich an, in Break-Out-Räume zu gehen. Dafür werden mehrere Mentor*innen benötigt, die wissen, wie man die einzelnen Projekte baut. Diese können dann eine kleinere Gruppe Teilnehmer*innen in einem Break-Out-Raum betreuen, wodurch alle eine angemessene Chance auf Hilfestellung haben. Wir empfehlen vier bis fünf Teilnehmer*innen je Break-Out-Raum. So kann man auch leichter auf unterschiedliche Wissensstände oder verschieden schnelles Arbeitstempo eingehen.

Genutzte Links:

Jugend hackt, pädagogisches Konzept:

<https://jugendhackt.org/blog/unser-verhaltenskodex-code-of-conduct-und-unser-aushang-deine-rechte/>

Was ist eigentlich Scratch?: <https://www.youtube.com/watch?v=SUSSsZLxjDs&t=182s>

Scratch-Plattform: <https://scratch.mit.edu/>

Einführung Scratch Entwicklungsumgebung 1: <https://www.youtube.com/watch?v=jYbTamyEXxI>

Einführung Scratch Entwicklungsumgebung 2:

<https://scratch.mit.edu/projects/editor/?tutorial=getStarted>

Beispielprojekt "Pong-Spiel": <https://scratch.mit.edu/projects/493513953/>

Tutorial "Jumping Game": <https://www.youtube.com/watch?v=1jHvXakt1qw>