InfoSphere
World of Informatics

http://schuelerlabor.informatik.rwth-aachen.de

Spielend Programmieren lernen mit Scratch

Verfasser: Evelyn Willam

Kurz-Info:

Informatischer Inhalt: Grafische Programmierung mit Scratch

Jahrgangsstufe: 6 bis 8

Vorwissen: Keins

CC BY SA

KURZINFORMATION FÜR DIE LEHRKRAFT

Titel: *Spielend Programmieren lernen mit Scratch*

Schulstufe: *Unter- und Mittelstufe*

optimale Jahrgangsstufe: *Klasse 6 bis 8*

Themenbereich: *Grafische Programmierung, Variablen, Kontrollstrukturen, Bildbearbeitung*

EINORDNUNG IN GESETZLICHE RAHMENBEDINGUNGEN

Bildungsstandards der GI: ***Modellieren und Implementieren, Begründen und Bewerten, Strukturieren und Vernetzen, Kommunizieren und Kooperieren, Algorithmen, Informatiksysteme***

Einbindung in den Unterricht: Das Modul biete sich ideal an, um die Schülerinnen und Schüler mit viel Freude an das Programmieren heranzuführen. Hierzu wird die grafische Programmierungsumgebung des MIT Scratch Editors (http://scratch.mit.edu) genutzt, die vorgefertigte Programmierelemente in Form von Puzzleteilen bereitstellt, welche dann entsprechend zusammengesetzt werden können.

Vorkenntnisse: keine

Dauer: *ca. 5 Stunden*

Inhaltsbeschreibung: Zu Beginn werden den Schülerinnen und Schülern die Grundelemente und die Handhabung der Entwicklungsumgebung Scratch Editor vorgestellt.

In der ersten Phase eines Gruppenpuzzles werden die Schülerinnen und Schüler in Zweier-Teams eingeteilt, in denen sie anhand von kleinschrittigen Anleitung in Form von Arbeitsblättern „Expertenwissen“ zu einem von vier Grundlagenschwerpunkten (Bereichen) sammeln, wie z.B. Variablen.

Aufbauend darauf arbeiten am Ende Experten aus jedem der vier Bereiche zusammen um ein kleines, fantasievolles Spiel zu kreieren.

INHALT

Kurzinformation für die Lehrkraft 2
Lernziele 4
Einordnung in Gesetzliche Rahmenbedingungen 4
Bildungsstandards der GI 4
Einbettung in den Schulunterricht 5
Didaktische/Methodische Schwerpunktsetzung 6
Verlaufsplan des Moduls 8
Quellenverzeichnis 9
Abbilddungsverzeichnis 9
Anhang 10

LERNZIELE

Die Schülerinnen und Schüler...

- verstehen grundlegende Konzepte der Schleifen und Kontrollstrukturen.
- verstehen grundlegende Prinzipien des Projektmanagements (in Form eines Gruppenpuzzles) und setzen diese um.
- bedienen sich simpler Methoden, um mit dem Computer zu interagieren.
- verstehen Programme und sind fähig diese zu analysieren.
- verwenden die Umgebung Scratch mit ihren Funktionen zur Erstellung eines Programms.
- präsentieren ihre Ergebnisse.

EINORDNUNG IN GESETZLICHE RAHMENBEDINGUNGEN

BILDUNGSSTANDARDS DER GI

- Modellieren und Implementieren
    - Implementieren Modell mit geeigneten Werkzeugen
    - Reflektieren Modelle und deren Implementierung, bzw. Modellkritik
        - Überprüfung der Angemessenheit der Lösung
        - Bewertung der erreichten Resultate/Implementierung

- Begründen und Bewerten
    - Begründen Entscheidungen bei der Nutzung von Informatiksystemen *(Schüler-innen begründen bei der Präsentation ihrer Ergebnisse ihr Vorgehen)*
        - Nennen Vor- und Nachteile
        - Können Argumente nachvollziehen
        - Begründen die Darstellung

- Strukturieren und Vernetzen
    - Strukturieren Sachverhalte durch zweckdienliches Zerlegen und Anordnen *(Schüler-innen zerlegen die Spielprogrammierung in einzelne Schritte, welche teils durch die Arbeitsmaterialien vorgegeben sind)*
        - Zerlegen Sachverhalte durch Erkennen und Abgrenzen von einzelnen Bestandteilen
        - Erkennen Reihenfolgen in Handlungsabläufen

- Kommunizieren und Kooperieren *(in Teams und Gruppenarbeit)*
    - Kommunizieren fachgerecht über informatische Sachverhalte
        - Tauschen sich untereinander verständlich über informatische Inhalte aus
        - Stellen informatische Sachverhalte unter Benutzung von Fachbegriffen mündlich und schriftlich sachgerecht dar *(denn später müssen die Schüler-innen ihr Vorgehen präsentieren und begründen)*
    - Kooperieren bei der Lösung informatischer Probleme
        - Kooperieren in verschiedenen Formen der Zusammenarbeit bei der Bearbeitung einfacher informatischer Probleme
        - Kooperieren in arbeitsteiliger Gruppenarbeit

        - Beschreiben die Bearbeitung und Ergebnisse in einem gemeinsamen Dokument *(durch Gruppenpräsentation am Ende)*

- Algorithmen
    - Kennen Algorithmen zum Lösen von Aufgaben und Problemen aus verschiedenen Anwendungsgebieten und lesen und interpretieren gegebene Algorithmen *(durch Arbeitsblätter)*
        - Lesen und verstehen Handlungsvorschriften für das Arbeiten mit Informatiksystemen
        - Interpretieren Handlungsvorschriften korrekt und führen sie schrittweise aus
    - Entwerfen und realisieren Algorithmen mit den algorithmischen Grundbausteinen und stellen diese geeignet dar *(Befehlsblöcke von Scratch)*
    - Benutzen die algorithmischen Grundbausteine zur Darstellung von Handlungsvorschriften *(Befehlsblöcke von Scratch)*
        - Entwerfen Handlungsvorschriften als Text oder mit formalen Darstellungsformen *(formale Darstellung durch das Zusammensetzen der Bausteine von Scratch)*
        - Entwerfen und testen einfache Algorithmen *(Schüler-innen erstellen zuerst kleinere Programme und schließlich das Spiel)*

- Informatiksysteme
    - Wenden Informatiksysteme zielgerichtet an
        - Arbeiten mit grafischen Benutzungsoberflächen *(Scratch)*

EINBETTUNG IN DEN SCHULUNTERRICHT

Dieses Modul kann als Einstieg in die Programmierung genutzt werden. Im weiteren Verlauf kann auf andere grafische Programmierumgebungen wie z.B. den „MIT App Inventor“ (http://appinventor.mit.edu/) zurückgegriffen werden, zudem ebenfalls InfoSphere-Module angeboten werden.

DIDAKTISCHE/METHODISCHE SCHWERPUNKTSETZUNG

Beschreibung des Modulablaufs

Phase 1

Zu Beginn des Moduls wird den Schülerinnen und Schülern der **Scratch online Editor** präsentiert, wobei die Programmieroberfläche mit ihren einzelnen Komponenten und Funktionen, inklusive ihrer visuellen Programmiersprache, durch Beispiele anschaulich gemacht wird. Des Weiteren werden die Schülerinnen und Schüler davon in Kenntnis gesetzt, dass es sich bei dem Modul um ein **Gruppenpuzzle** handelt und sie zunächst zu einem von vier möglichen Experten (BZ, GA, KK, FV)[1] „ausgebildet" werden, um nach Abschluss der Ausbildungsphase zu einer entsprechenden, aus den vier Kategorien bestehenden Expertengruppe zusammengesetzt werden, welche zum Ziel hat ein Fangspiel zu entwickeln.

Phase 2

Nach einer Pause werden die Schülerinnen und Schüler in **Experten-Ausbildungsgruppen** eingeteilt, bevor sie mit der Bearbeitung der **ersten Arbeitsblätter in Partnerarbeit** beginnen. Durch die kleinschrittigen Anleitungen in den Arbeitsblättern lernen sie alle Grundlagen, die sie als ausgebildete Experten benötigen, um in der späteren Gruppenphase sicher, anhand eines weiteren Arbeitsblattes, das bisher Gelernte anwenden und umsetzen können. Zu beachten in dieser Phase ist, dass idealerweise alle Schülerinnen und Schüler einer Expertengruppen am selben Tisch sitzen, sodass sich die Zweier-Gruppen auch untereinander austauschen und unterstützen können. Eine darauf folgende **Sicherungsphase**, zu der sowohl die Sicherungsfragen des jeweiligen Arbeitsblattes, als auch eine individuelle Besprechung des bisher erstellten Programmes mit einer betreuenden Person gehören, festigt den neu erlernten Stoff. Hierbei sollte darauf geachtet werden, dass sich alle Schülerinnen und Schüler auf einem soliden Wissensstand befinden, so dass sie sich in ihrer kommenden Gruppe gut als Experte einbringen können und ihren Wissensstand entsprechend teilen – denn nur ein Experte innerhalb der Vierer-Gruppe, verfügt nur über den Wissensstand von einem Themenbereich und trägt damit eine hohe Verantwortung.

Phase 3

Nach einer weiteren Pause und einem Warmup-Spiel, welches den Wiedereinstieg erleichtern soll, werden Gruppen mit jeweils einem Experten aus jedem Expertenbereich gebildet (Gruppenpuzzleprinzip), die schließlich anhand eines weiteren Arbeitsblattes nach dem *Prinzip der abnehmenden Hilfe* ein **Fangspiel** erstellen. Die hierbei geschaffenen Resultate werden am Ende des Workshops von den Entwicklerteams mithilfe des Smartboards **präsentiert**, bevor eine **Feedbackrunde** den Workshop abschließt.

[1] Arbeitsblätter Bewegung und Zeichnen (BZ), Grafik und Aussehen (GA), Klang und Kommunikation (KK) und Fühlen und Variablen (FV)

Begründung des Methoden- und Medieneinsatzes

Im InfoSphere - Schülerlabor Informatik ist eines der wichtigsten Prinzipien, dass die Schülerinnen und Schüler selbstständig arbeiten um informatische Probleme zu lösen. Dabei sollen sie in Teams möglichst ohne direkte Anleitung durch eine Lehrperson arbeiten und sich gegenseitig unterstützen. Um dies zu gewährleisten wurden für dieses Modul Arbeitsblätter entwickelt, die die Schülerinnen und Schüler durch die einzelnen Phasen führen sollen. Dabei ist die Rolle der Betreuerinnen und Betreuer eine rein beratende. Um trotzdem den zeitlichen Rahmen und vor allem das Erfolgserlebnis der Teilnehmerinnen und Teilnehmer gewährleisten zu können, wird auf die kurze Einstiegspräsentation der Programmieroberfläche zurückgegriffen. Bei der Programmieroberfläche Scratch handelt es sich um eine Drag&Drop-Oberfläche, die zudem ihre eigene Programmiersprache, mit selbigem Namen stellt.

Didaktische Prinzipien nach Baumann und Hubwieser

Da sich dieses Modul nach dem Konzept des Schülerlabors InfoSphere richtet, liegt ein besonderer Fokus auf dem *Prinzip des aktiven Lernens*. Durch die Arbeitsmaterialien ist ein selbstständiges und aktives Lernen der Schülerinnen und Schüler gegeben. Sie müssen selber aktiv werden, um Probleme zu lösen. Während der Begrüßung der Schülerinnen und Schüler (*Phase 1*) wird auch auf den Ablauf des Moduls eingegangen. Hier wird ihnen das Ziel des Moduls mitgeteilt. Auf den Arbeitsmaterialien (*Phase 2*) wird in einem kurzen Einleitungstext das Teilziel des jeweiligen Arbeitsblattes erläutert. Hierdurch wird das *Prinzip der Zielvorstellung* erfüllt.

Durch die relativ freie Aufgabenstellung in der zweiten Erarbeitungsphase (*Phase 3*), der Gestaltung des Fangspiels und der Möglichkeit bestimmte Teile des Fangspiels auszulassen bzw. als Bonus zusätzlich zu integrieren, wird auf das *individuelle Lerntempo* der Schülerinnen und Schüler eingegangen. Diese können bei Bedarf, neben dem verlangten Pflichtteil des Fangspiels individuell dessen Komplexität erhöhen oder sich bei Bedarf bei den betreuenden Personen Hilfe holen.

VERLAUFSPLAN DES MODULS

Zeit	Phase	Inhalt	Medium	Sozialform
0:00-0:05	Begrüßung	*Vorstellung InfoSphere und Betreuerteam*		LV
0:05-0:25	Vorstellung Modul	*Vorstellung Oberfläche, Experteneinteilung*	Smartboard/Beamer, PC, Leitfaden	LV
0:25-1:40	Expertenausbildung	*Erstellung eines ersten Programms (gemäß jeweiligem Arbeitsblatt)*	AB, Gruppentisch pro Expertenarbeitsblatt	PA, GA
1:40-1:50	Sicherungsphase	*Besprechung der Sicherungsfragen oder persönliche Besprechung des bisherigen Programms/Programmteils (jeweils tisch- bzw. teamweise)*	AB	D
1:50-2:05	Pause			
2:05-2:15	Warmup-Spiel			
2:15-2:20	Gruppeneinteilung	*Einteilung in Expertengruppen*	Expertenkarten	LV
2:20-3:55	Spielprogrammierung	*Programmierung in Expertengruppen*	AB	GA
3:55-4:05	Pause			
4:05-4:45	Präsentation	*Vorstellung der Ergebnisse*	Smartboard/Beamer	SV
4:45-5:00	Verabschiedung	*(u.U. Urkundenverleihung) Feedbackrunde*		UG

AB – Arbeitsblatt, LV – Lehrervortag, UG – Unterrichtsgespräch, EA – Einzelarbeit, PA – Partnerarbeit, GA – Gruppenarbeit, SV – Schülervortrag, D - Diskussion

QUELLENVERZEICHNIS

- http://scratch.mit.edu
- Gesellschaft für Informatik e.V. (GI) (Hg.) (2008): Grundsätze und Standards für die Informatik in der Schule. Bildungsstandards Informatik für die Sekundarstufe I. Gesellschaft für Informatik e.V. (GI). Online verfügbar unter http://www.fa-ibs.gi-ev.de/fileadmin/gliederungen/fb-iad/fa-ibs/Empfehlungen/bildungsstandards_2008.pdf, zuletzt geprüft am 29.06.2015.

ABBILDDUNGSVERZEICHNIS

- Alle Grafiken entstammen der Software Scratch, welche von dem MIT Media Lab entwickelt wurde.
- Scratch ist lizenziert unter Creative Commons Attribution.ShareAlike 2.0.

ANHANG

- Arbeitsblätter (auf Anfrage: in editierbarer Form)
- Musterlösung für Betreuer/innen

Der Online-Editor

Du willst das Programm SCRATCH von zu Hause aus nutzen, weißt aber nicht so recht wie?

1. Schritt: Öffne deinen Browser und gebe oben folgende URL ein: https://scratch.mit.edu/

2. Schritt: Damit du deine Projekte speichern kannst, solltest du dich kostenlos anmelden. Klick dazu einfach auf „Scratcher werden“.

Scratcher werden

3. Schritt: Wähle deinen Benutzernamen und überlege dir ein Kennwort, das du dir gut merken kannst. Fülle die Felder aus und klicke auf weiter.

4. Schritt: Jetzt möchte Scratch noch einige Informationen von dir, um ihre Statistiken zu erstellen. Gib also dein Geburtsdatum, Geschlecht und Heimatland (Deutschland=Germany) an.

5. Schritt: Am Schluss musst du noch eine Email-Adresse angeben, um einen Bestätigungslink zu erhalten. Nachdem du deinen Account über diesen Link aktiviert hast kannst du deine Projekte auch speichern.

6. Schritt: Los geht‘s!!! Klicke oben links auf Entwickeln. Und schon kannst du anfangen dein eigenes Spiel zu programmieren.

Viel Spaß!!!

CC BY SA

Scratch - Gruppenpuzzle -

Bewegung und Zeichnen

Während des gesamten Moduls führen euch die Arbeitsblätter durch die Programmerstellung und geben euch Hinweise.

Achtet dabei einfach auf folgende Symbole, die...

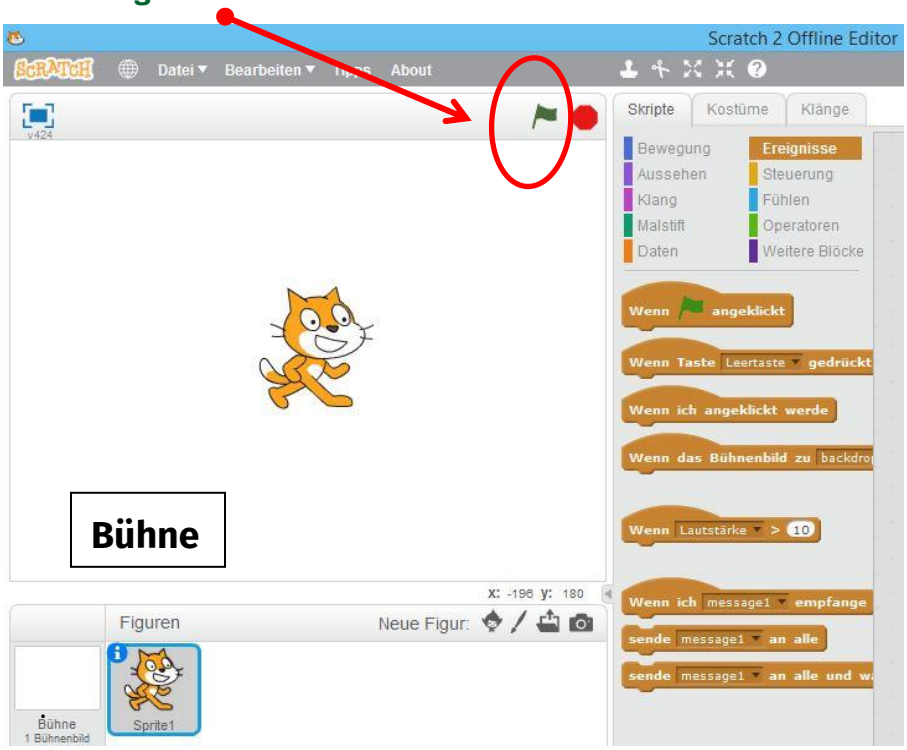
- euer Arbeiten strukturieren und Teilziele aufzeigen,
- euch Hilfen geben, Wichtiges, Schwieriges, etc. kennzeichnen und erklären
- Arbeitsaufträge und Aktionen beinhalten!

Gruppenpuzzle: Bewegung und Zeichnen

Mit dem Programm SCRATCH lernt ihr wie ihr **Figuren** (im Beispiel die Katze) so programmiert, dass sie sich bewegen und sogar **Aktionen** wie zeichnen ausführen.
Neugierig? Na dann los!

Lasst die Katze laufen:

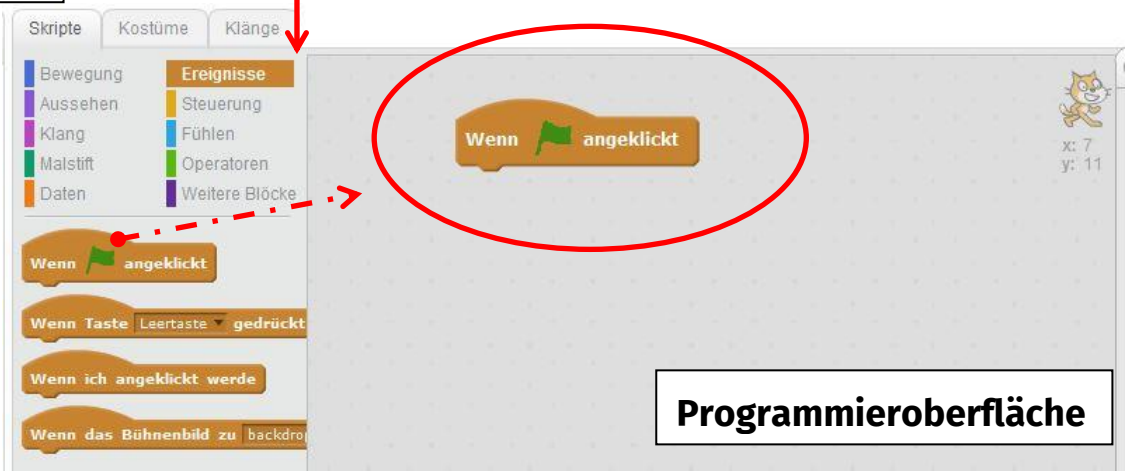
In unserem Programm soll die Katze immer genau dann laufen, wenn ihr mit der Maus das **grüne Fähnchen** über der **Bühne** anklickt.



CC BY SA

Dazu legen wir jetzt die **Startbedingung** fest. Hierzu wählt ihr im Menü „Skripte“ aus den **Ereignissen** die Startbedingung *„Wenn-grüne-Fahne-angeklickt“* und zieht sie einfach rechts daneben, in das freie Feld, eure **Programmieroberfläche**.

Menü

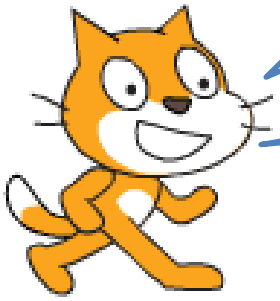


Nun wählt ihr im Menü *„Skripte“* aus der **Bewegung** die **Funktion** *„gehe-10er-Schritt“* aus und zieht sie genau unter eure Startbedingung (so dass die Puzzleteile ineinander einrasten).

Testet euer **Programm**, indem ihr über der Bühne *das grüne Fähnchen* anklickt.
Ihr könnt die Katze , nachdem sie gelaufen ist, auch immer wieder auf eine beliebige Position innerhalb der Bühne ziehen.

CC BY SA

Bewegung und Zeichnen



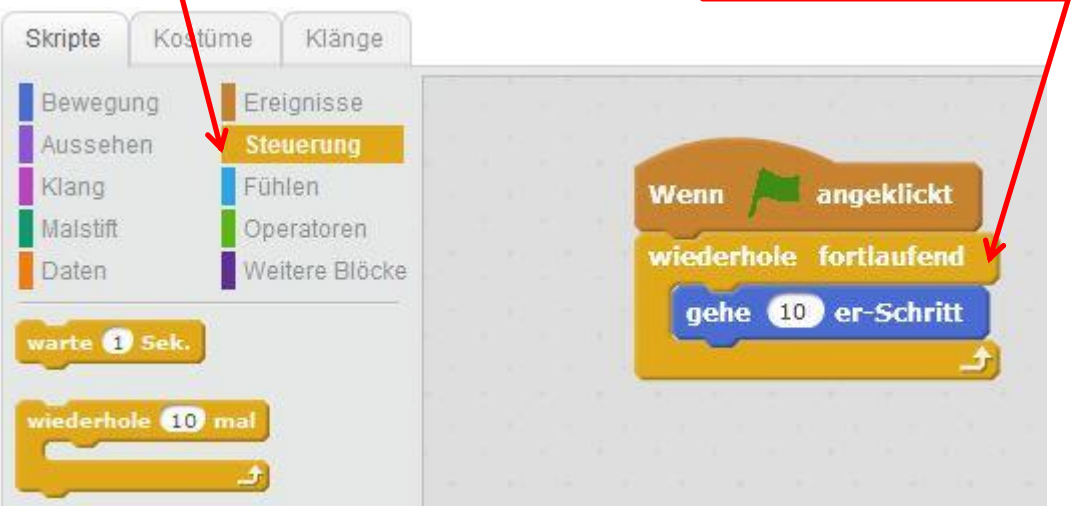
Was müsst ihr tun, wenn ich mehr oder weniger als in *10-er Schritten* vorwärts laufen soll?

Könnt ihr mich auch anders in **Bewegung** setzen?

- Schreibt es auf:

!
- Testet euer Programm anschließend .

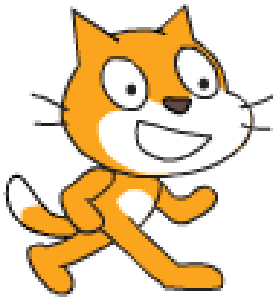
Jetzt lasst die auf der **Bühne** ganz nach rechts bis zum Rand laufen.
Damit ihr euer Programm nicht solange ausführen müsst, bis die Katze den Rand erreicht, verwendet hierfür eine **Schleife**.
Wählt hierzu aus der **Steuerung** das „wiederhole-*fortlaufend*“ aus und zieht es unter die **Startbedingung**.



CC BY SA

Scratch - Gruppenpuzzle -

Bewegung und Zeichnen



Wie lange laufe ich jetzt in **10er Schritten**?

Stoppe ich am Rand der Bühne automatisch?
Wie schafft ihr es, dass ich am Rand stoppe?

- Was denkt ihr? Notiert eure Überlegungen hier:

Wenn die jetzt den Rand der **Bühne** berührt, soll sie ***abprallen,*** die Richtung wechseln und weiter laufen.

- Überlegt mit welchen Puzzleteilen aus der **Bewegung** ihr das schaffen könnt.
- Setzt eure Überlegungen in die Tat um und testet euer Programm .

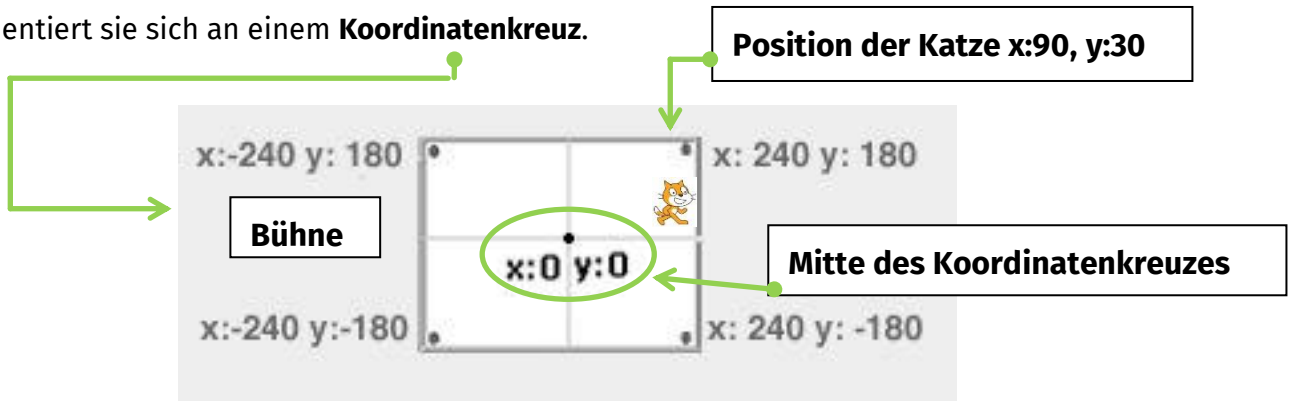
Die soll jetzt mit den **Pfeiltasten** bewegt werden.
Erstellt dazu ein neues Programm!
<u>Aufgepasst</u>: Jetzt müssen wir auch die **Startbedingung** ändern!

- Sucht euch die passende Startbedingung aus **Ereignisse**.
- dann die zwei zu unserem Ziel passenden Puzzleteile aus **Bewegung**.

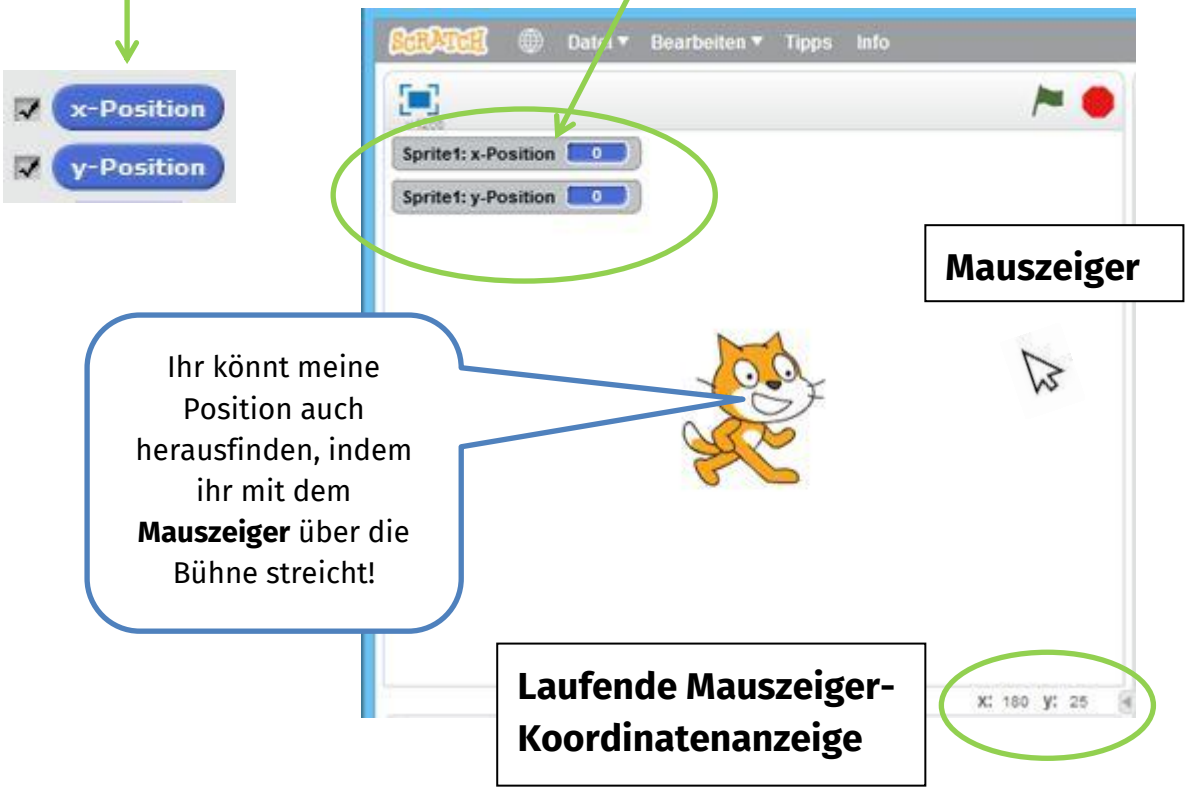
CC BY SA

Bewegung und Zeichnen

Die kann auch zu einer bestimmten von euch gewählten Stelle auf der Bühne laufen. Dabei orientiert sie sich an einem **Koordinatenkreuz**.



Um herauszufinden wo sich eure auf der Bühne jetzt genau befindet, könnt ihr aus der **Bewegung** die Koordinaten-Positionen anzeigen lassen. Dazu einfach anklicken.



CC BY SA

Bewegung und Zeichnen

1. Lasst die nach **rechts** oder nach **links** laufen

2. Bei welchen **Koordinaten** bleibt sie stehen? Notiert es:

x: _____

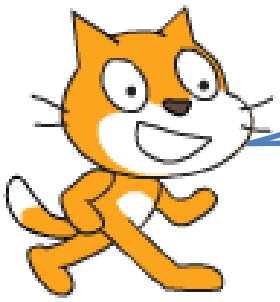
y:

3. Wählt eine beliebige **x-** und eine beliebige **y-Koordinate** auf der Bühne aus und lasst eure dorthin laufen. Nutzt dazu unterschiedliche Befehle aus der **Bewegung**.

4. Was bedeutet der Begriff **Koordinate**? Erklärt ihn:

Zur Erinnerung:
Alles was ihr in die **Schleife** zieht wird dauerhaft ausgeführt -also ohne Pause ständig wiederholt.

Die **Schleife** könnt ihr während ihrer Ausführung, durch klicken auf den roten Punkt über eurer Bühne stoppen.



Gut gemacht!
Kurz verschnaufen und auf zum Endspurt!!

CC BY SA

Lasst die Katze auf- und abtauchen:

Falls die den Rand der Bühne berührt, soll sie auf der gegenüberliegenden Seite wieder auftauchen und weiter laufen. Also nicht mehr mit dem Rand zusammen stoßen, oder über den Rand hinaus laufen.

Die zu diesem Ziel passende **Anweisung** findet ihr in der **Steuerung**:

Falls die Katze am rechten Rand angekommen ist, **dann** soll sie am linken Rand wieder auftauchen und weiter laufen.

„falls-dann“- **Anweisung**

Da wir verhindern wollen, dass die Katze über den Rand hinaus läuft, **vergleichen** wir jetzt ihre **aktuelle Position** (x-Position) mit der Position vom rechten Rand der Bühne.

Wisst ihr noch wie groß die **x-Koordinate** dort war?

Für diesen **Vergleich** bildet der *„größer-als-Operator“* das Grundgerüst.

Ihr findet ihn unter **Operatoren**.
Dort hinein zieht ihr jetzt die **x-Position** aus der **Bewegung** und schreibt die **x-Koordinate** des rechten Randes hinter das **>** -Zeichen.

Aus der **Bewegung** benötigt ihr jetzt noch diese Zuweisung: setze x auf -240

Damit sagt ihr der , dass sie am linken Rand (**x: -240**) wieder auftauchen soll.

- Lasst die jetzt gegen den rechten Rand eurer Bühne laufen und an ihrem linken Rand wieder auftauchen.
- Verwendet dazu **ZUSÄTZLICH**, zu den euch schon bekannten Befehlen, die oben erwähnten Befehle aus den Bereichen **Steuerung**, **Operatoren** und **Bewegung**.

CC BY SA

Jetzt wird`s kreativ:

Um verfolgen zu können, welchen Weg die gelaufen ist, könnt ihr den gelaufenen Weg nachzeichnen lassen.

Dazu wählt ihr aus dem **Malstift** einfach *„schalte-Stift-ein“* aus und zieht ihn in euer Programm. Die *Stiftdicke* und die *Stiftfarbe* könnt ihr selbst bestimmen. Komplizierter wird`s, wenn ihr versucht die Katze z. B. eine Blume zeichnen zu lassen..
Probiert es nach euren Vorstellungen aus!

Herzlichen Glückwunsch!
Ihr habt euer Ziel erreicht!

CC BY SA

Scratch - Gruppenpuzzle -

Fühlen und Variablen

Während des gesamten Moduls führen euch die Arbeitsblätter durch die Programmerstellung und geben euch Hinweise.

Achtet dabei einfach auf folgende Symbole, die..

- euer Arbeiten strukturieren und Teilziele aufzeigen,
- euch Hilfen geben, Wichtiges, Schwieriges, etc. kennzeichnen und erklären
- Arbeitsaufträge und Aktionen beinhalten!

Gruppenpuzzle: Fühlen und Variablen

Hier lernt ihr mit dem Programm SCRATCH, wie ihr *Figuren* (im Beispiel die Katze) beibringt etwas zu **fühlen** und wofür man in diesem Zusammenhang die **Variablen** braucht.
Neugierig? Na dann los!

Fühlen mit Hilfe von Variablen:

Die Katze soll zunächst warten bis sie berührt wird und daraufhin nach eurem Namen fragen, den ihr dann auch eintragt! Das Ganze jetzt Schritt für Schritt ☺

Dafür legen wir erst einmal die **Startbedingung** fest. Hierzu wählt ihr im Menü „Skripte“ aus den **Ereignissen** die Startbedingung *„Wenn-grüne-Fahne-angeklickt“* und zieht sie einfach rechts daneben, in das freie Feld, eure **Programmieroberfläche**.

CC BY SA

Scratch - Gruppenpuzzle -

Fühlen und Variablen

Die soll auf euch aufmerksam werden, sobald ihr sie mit dem **Mauszeiger berührt**.

Dazu sucht ihr euch das folgende Puzzleteil unter dem Menüpunkt *Skripte* im Bereich

Steuerung:

Aufgrund dieses Puzzleteils **wartet** die solange **bis** **eine Bedingung** –die ihr noch in das Puzzleteil einfügen könnt- **erfüllt ist**. In unserem Beispiel **wartet** sie also so lange, **bis** ihr sie mit dem **Mauszeiger berührt**.

Hierfür verwendet ihr folgende **Bedingung**, die ihr unter **Fühlen** findet:

Hier könnt ihr den Mauszeiger auswählen.

Sobald ihr die berührt habt, soll sie jetzt **die Frage** stellen, wer sie denn streicheln möchte. Verändert dazu das folgende Puzzleteil aus **Fühlen** dementsprechend:

Ihr könnt hier in das Puzzleteil die Frage hineinschreiben.

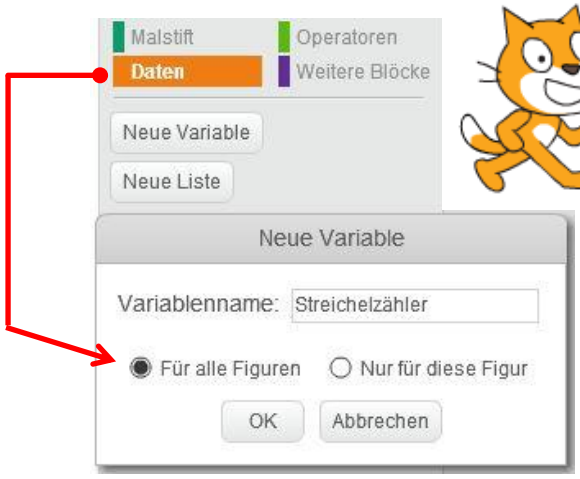
So soll euer Programm bisher aussehen.

Hier könnt ihr jetzt euren Namen eintragen!

CC BY SA

Fühlen und Variablen

Damit sich die auch *merkt*, wie oft ihr sie gestreichelt habt (also angeklickt habt) legen wir jetzt einen *„Merkspeicher“* an, eine sogenannte **Variable.** Wir geben ihr einen passenden Namen, wie z.B. *„Streichelzähler“*.



Durch Häkchen setzen wird der *Streichelzähler* - eure Variable - auf der Bühne sichtbar!

Jedes Mal wenn wir auf die grüne klicken, soll die wieder bei „0“ anfangen zu zählen. Deshalb setzen wir in diesem Programmteil **die Variable** auf 0, so weiss die ab wann sie anfagen soll zu zählen.

Damit dies in eurem Programm funktioniert, **setzt ihr den Streichelzähler auf 0**. Das passende Puzzleteil findet ihr unter **Daten**. Zieht es einfach unter eure vorhandenen Puzzleteile in eure Programmieroberfläche.

Wenn euer Programm so aussieht, habt ihr alles richtig gemacht! Auf zum zweiten Teil!

CC BY SA

Fühlen und Variablen

Was genau tut euer Programm bisher? Was kann ich **fühlen**? Was macht die **Variable**? Erklärt den bisherigen Programmablauf mit euren eigenen Worten!

1.

2.

3.

4.

Zur Erinnerung:
Wir wollen dass die Katze merkt, dass wir sie streicheln und ab wann wir dies tun.
Zusätzlich lassen wir unsere Zählungen auf der Bühne mit Hilfe unserer erstellten Variablen (alias „Streichelzähler“) anzeigen.

CC BY SA

Zählen mit Hilfe von Variablen:

Nachdem wir unsere **Variable** (den Merkspeicher) erstellt haben, wollen wir ihn jetzt auch benutzen. Die Katze soll jetzt zählen, bzw. sich merken, wie oft sie gestreichelt wurde. Das machen wir in einem **zweiten Programmteil**, den wir mit der **Startbedingung** *„Wenn-ich-angeklickt-werde“* beginnen.
Dieser Programmteil wird dann jedes Mal aufgerufen, wenn wir die Katze anklicken, also **berühren**.

Aus dem Menü Skripte wählt ihr aus den **Ereignissen** *„Wenn-ich-angeklickt-werde“*, aus. Und zieht es unter euer erstes Programm.

Zieht nun *„ändere-Streichelzähler-um-1“* aus **Daten** eurer erstellten **Variable** unter diesen Block, um die zählen zu lassen. Dies sollte jetzt wie folgt aussehen:

Als nächstes bringen wir der bei sich zu bedanken, sobald sie 5-mal gestreichelt wurde.
Hierfür verwenden wir die *„falls-dann“*- **Anweisung** aus der **Steuerung**.

Falls die Katze 5-mal gestreichelt wurde,
dann soll sie sich bedanken.

In die **Anweisung** zieht ihr jetzt noch die **Bedingung**, die erfüllt sein soll. In unserem Fall ist das, ***falls die Katze 5-mal gestreichelt worden ist***, also der **Streichelzähler gleich 5** ist.

Wir verwenden dazu den **Vergleichs-Operator**:

Er vergleicht, ob der linke Wert gleich dem Rechten ist.

Dies sieht dann so aus:

CC BY SA

Scratch - Gruppenpuzzle -

Fühlen und Variablen

Damit sich die bei euch fürs Streicheln bedankt, müssen wir in die **„falls – dann“ – Anweisung** nur noch dieses Puzzleteil einsetzen: sage . In das weiße Feld schreibt ihr dann: „Danke fürs Streicheln!“
Das Puzzleteil findet ihr unter **Aussehen**.
Klar soweit? ☺ Dann testet euer Programm!

Hat euer Programm funktioniert?

Wozu habt ihr im ersten Teil eigentlich euren Namen eingegeben?

- Genau, weil die Katze sich jetzt namentlich bei euch bedanken soll!
- Doch woher weiß die Katze euren Namen noch?

- Weil sie ihn in der bereits vorhandenen **Variable** Antwort gespeichert hat und deshalb jederzeit darauf zugreifen kann.

Verbindet deshalb jetzt noch schnell mit Hilfe eines **Operators** die Antwort mit dem Satz den ihr sagen wollt und dann sollte das Ganze so aussehen:

Sieht der zweite Programmteil so bei euch aus?

Herzlichen Glückwunsch!
Ihr habt euer Ziel erreicht!

CC BY SA

Scratch - Gruppenpuzzle -

Grafik und Aussehen

Während des gesamten Moduls führen euch die Arbeitsblätter durch die Programmerstellung und geben euch Hinweise.

Achtet dabei einfach auf folgende Symbole, die..

- euer Arbeiten strukturieren und Teilziele aufzeigen,
- euch Hilfen geben, Wichtiges, Schwieriges, etc. kennzeichnen und erklären
- Arbeitsaufträge und Aktionen beinhalten!

Gruppenpuzzle: Grafik und Steuerung

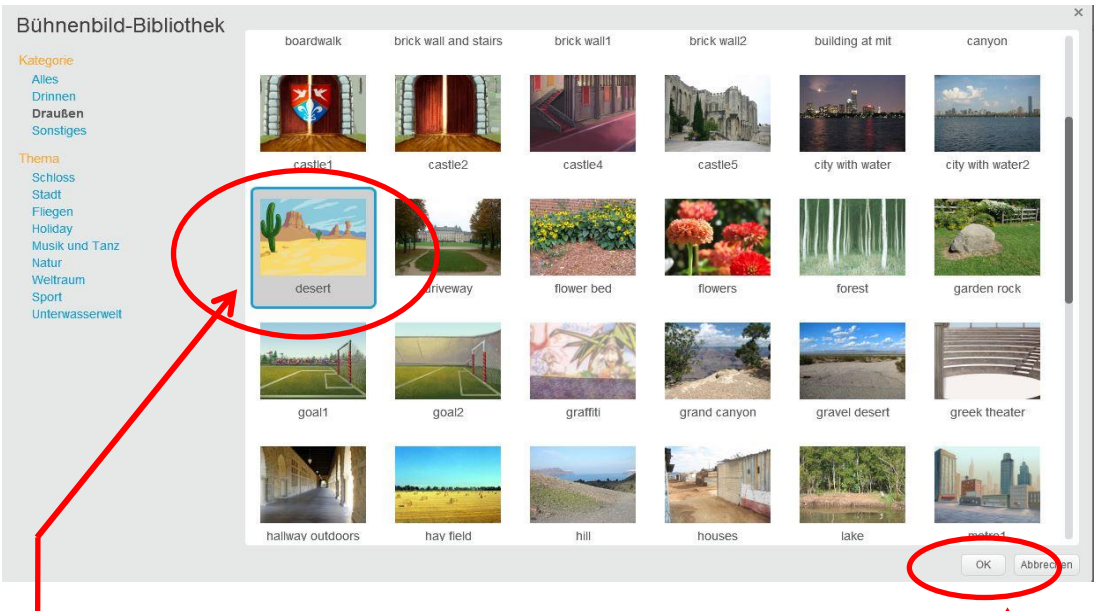
Hier lernt ihr mit dem Programm SCRATCH wie ihr **Hintergründe** anlegt und **Figuren** (im Beispiel die Katze) erstellt, die ihr dann mit Hilfe eurer Tastatur steuern könnt.
Neugierig? Na dann los!!

Hintergrund anlegen:

Dazu klickt ihr mit der Maus einfach auf diese Schaltfläche:

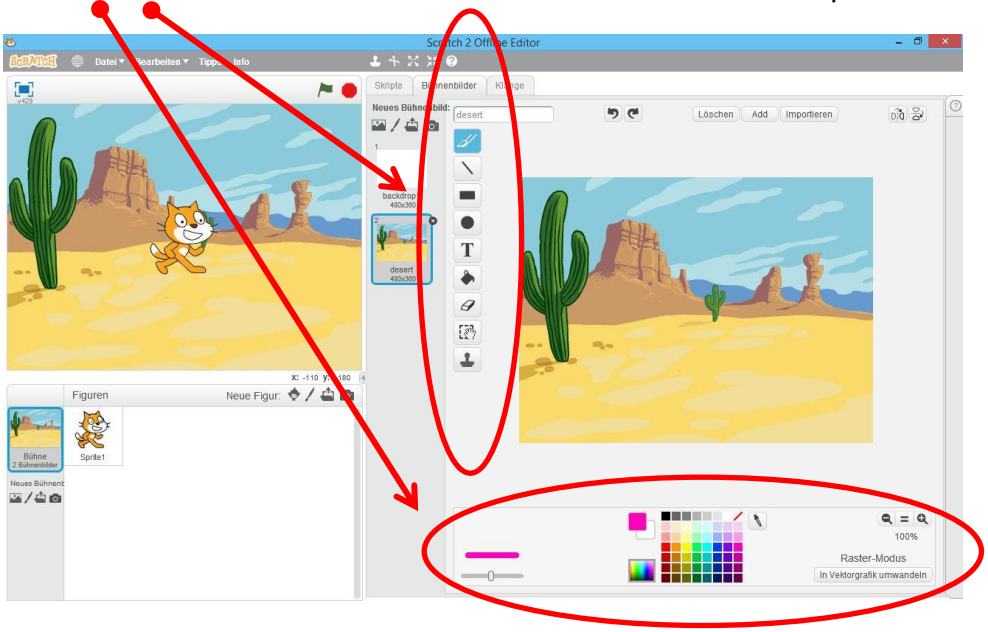
CC BY SA

Jetzt öffnet sich die **Bühnenbild-Bibliothek**.
Durchstöbert die Bühnenbild-Bibliothek und wählt euch jetzt einen Hintergrund durch Klick auf die linke Maustaste aus



Bestätigt eure Auswahl jetzt mit einem *Doppelklick*, oder klickt auf die Schaltfläche „***OK***“.

Ihr gelangt automatisch in den Bearbeitungsmodus und könnt jetzt mit den **Zeichentools** eurer Kreativität freien Lauf lassen - einfach ausprobieren!

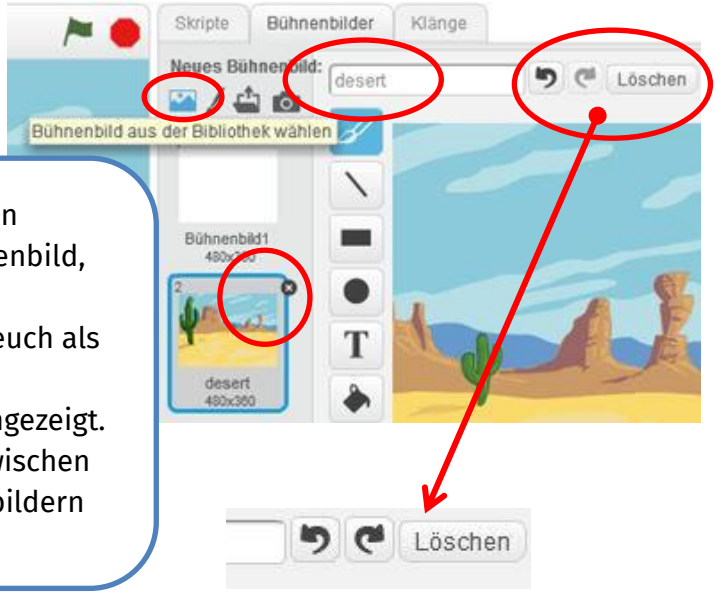


CC BY SA

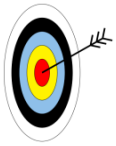
Grafik und Aussehen

In Menü unter **Bühnenbilder** werden all eure Bühnenbilder im Miniaturformat gezeigt. Ihr könnt sie hier noch einmal **umbenennen**, **löschen** oder auch weitere, ganz **neue** Bühnenbilder hinzufügen.

Klickt ihr auf ein Miniatur-Bühnenbild, so wird es **blau** umrahmt und euch als **Bühne** und zur Bearbeitung angezeigt. So könnt ihr zwischen euren Bühnenbildern wechseln.



Mit den **Pfeilen** könnt ihr eure Bühnenbildbearbeitung **rückgängig** machen oder **wiederherstellen**. Mit „Löschen“ löscht ihr alle Veränderungen inklusive Bild. ⚠



Figuren anlegen:

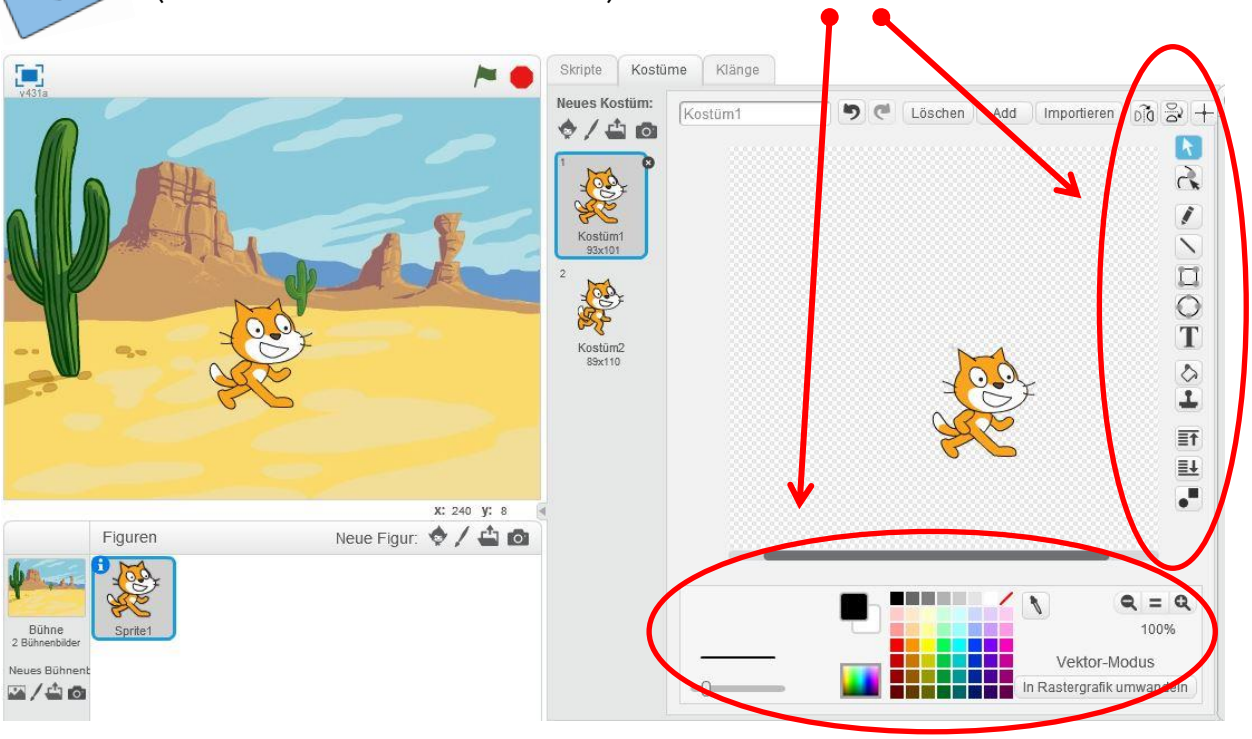
Passend zu euren neuen Bühnenbildern gilt es jetzt, neue **Figuren** anzulegen.

Dazu klickt ihr einfach auf die Schaltfläche *Neue Figur*. Wählt jetzt eine beliebige Figur aus der **Figurenbibliothek** aus.

CC BY SA

Grafik und Aussehen

Nachdem ihr eine Figur aus der **Figurenbibliothek** gewählt habt, könnt ihr sie (wie zuvor bei den Bühnenbildern) mit Hilfe der Zeichentools verändern.



Aufgepasst! Euer Menü verändert sich, je nachdem ob ihr ein Bühnenbild oder eine Figur bearbeiten wollt!

Ihr könnt zwischen den Menüs wechseln, in dem ihr unter der Bühne auf eure gewünschte Option klickt. Die Markierung zeigt dann an in welchem Menü ihr euch befindet.

Prima! Die erste Hälfte ist geschafft!
Kurz verschnaufen und weiter geht's!

CC BY SA

Kostümwechsel:

Eure Figur soll jetzt ihr Erscheinungsbild - das **Kostüm** - und damit auch gleichzeitig ihre **Körperhaltung** ändern.
Die dafür notwendigen Funktionen findet ihr in eurem Menü unter ***Kostüm*** sowie unter ***Skripte.***
Achtet darauf, dass eure Figur markiert ist, da sonst die Funktionen unter Skripte nicht verfügbar sind.

Legt einen Kostümwechsel an:

Zunächst wählt ihr im Menü unter **Kostüm** ein weiteres *Kostüm* und damit eine weitere *Körperhaltung* für eure Figur aus.

Anschließend geht ihr im Menü unter **Skripte**. Hier könnt ihr unter **Ereignisse** und **Aussehen** mit einfachen Funktionsanweisungen angeben, wann der Kostümwechsel beginnen soll und wie schnell der Wechsel abläuft.

CC BY SA

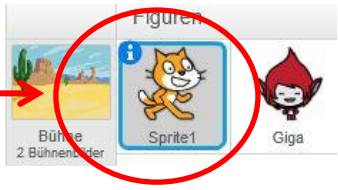
Scratch - Gruppenpuzzle -

Grafik und Aussehen

- Probiert andere Puzzelteile aus **Aussehen** aus.
- Experimentiert zusätzlich mit den Puzzleteilen aus **Steuerung**, z.B. mit:

- Erstellt je ein Programm für eure Figuren.
- Testet euer Programm .

Die Puzzleteile lassen sich nur aufrufen, wenn eure Figur markiert ist.



Meine Körperhaltung/ Kostüm könnt ihr jetzt ja schon ändern

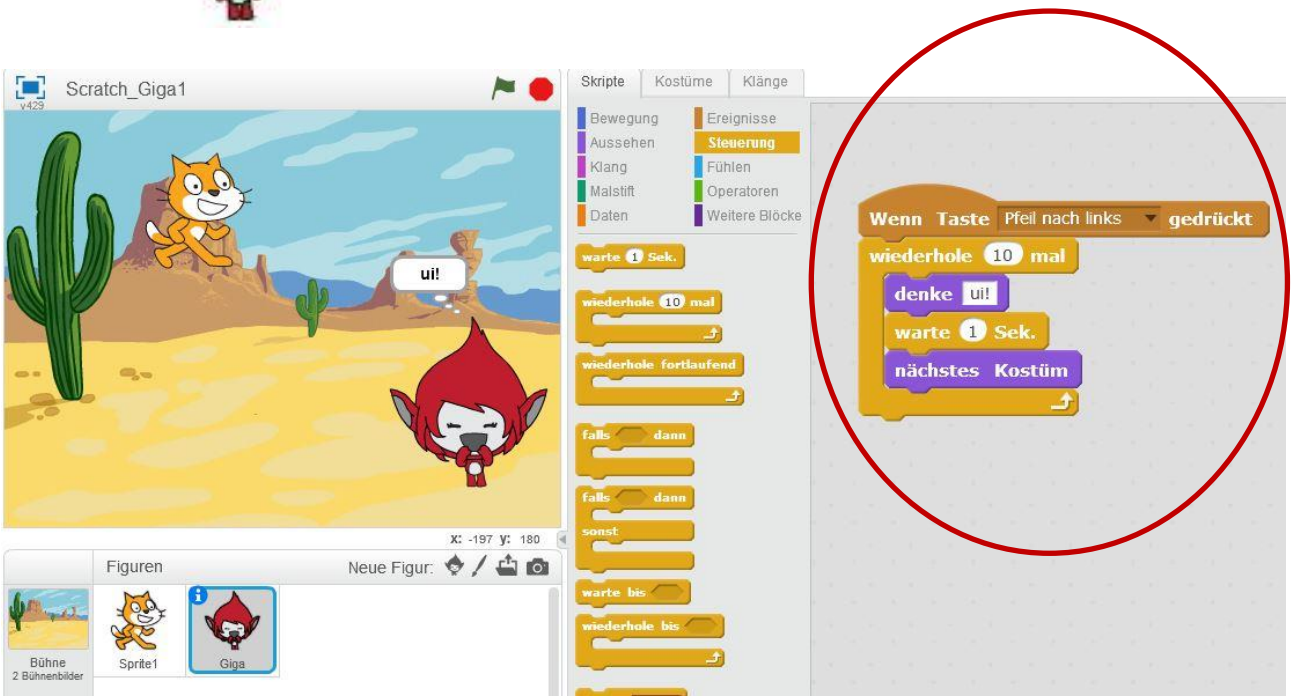
- Versucht jetzt mich oder einen meiner Freunde zum **Denken** zu bringen!
- Überlegt euch welche Bereiche aus **Skripte** ihr jetzt nutzen müsst!
- Erstellt dann ein neues Programm und testet es!

CC BY SA

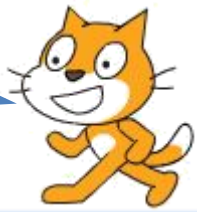
Scratch - Gruppenpuzzle -

Grafik und Aussehen

Habt ihr mich zum Denken gebracht? Oder einen meiner Freunde... ?



Was genau macht Giga laut dem Programm oben?
Gebt es mit euren eigenen Worten wieder!



CC BY SA

Vergesst nicht die grafischen Elemente wie z.B. Farbänderung, Größenänderung oder Formveränderung für eure Figuren zu nutzen, um individuelle Effekte zu erzielen.

Zusatzaufgabe (für alle die noch Puste haben) :

Ihr könnt euer Bühnenbild innerhalb eines Programmablaufs auch auswechseln.

Probiert es aus! Und falls ihr nicht drauf kommt, fragt eure Betreuer☺

Herzlichen Glückwunsch!
Ihr habt euer Ziel erreicht!

Scratch - Gruppenpuzzle -

Klang und Kommunikation

Während des gesamten Moduls führen euch die Arbeitsblätter durch die Programmerstellung und geben euch Hinweise.

Achtet dabei einfach auf folgende Symbole, die...

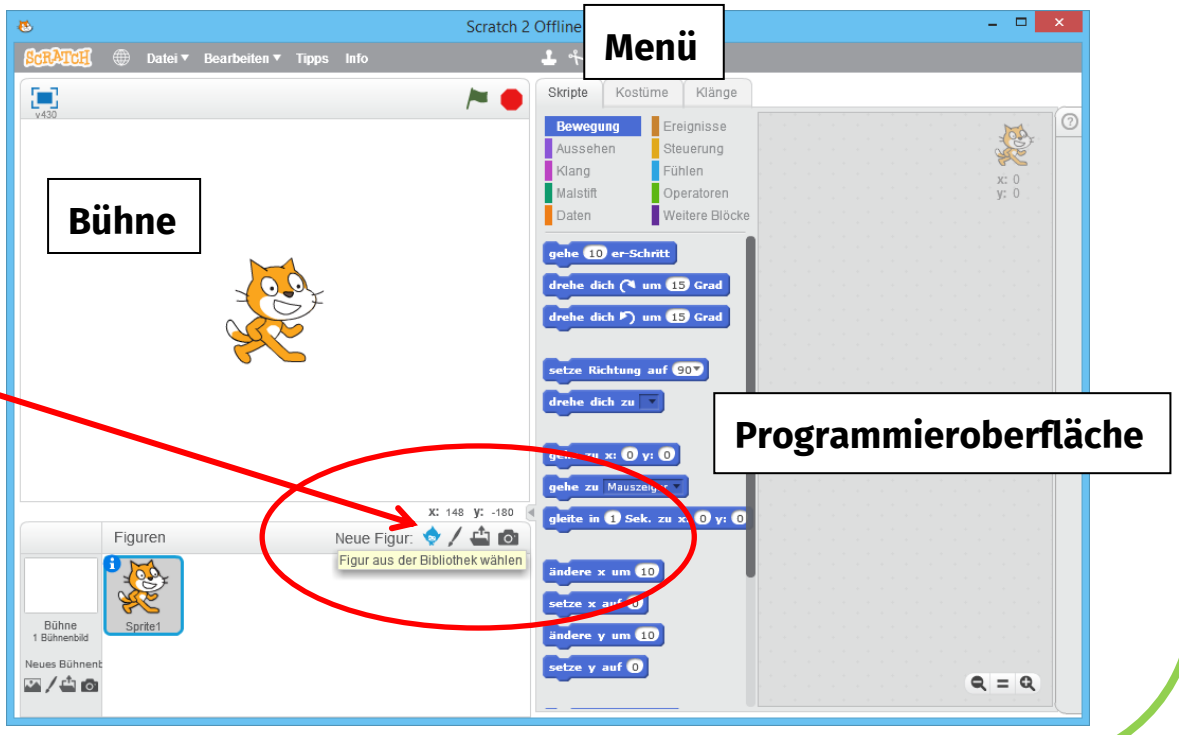
- euer Arbeiten strukturieren und Teilziele aufzeigen,
- euch Hilfen geben, Wichtiges, Schwieriges, etc. kennzeichnen und erklären
- Arbeitsaufträge und Aktionen beinhalten!

Gruppenpuzzle: Klang und Kommunikation

Hier lernt ihr wie ihr **Figuren** mit dem Programm SCRATCH erstellt (im Beispiel die Katze) und wie diese untereinander kommunizieren können. Neugierig? Na dann los!

Zweite Figur anlegen:

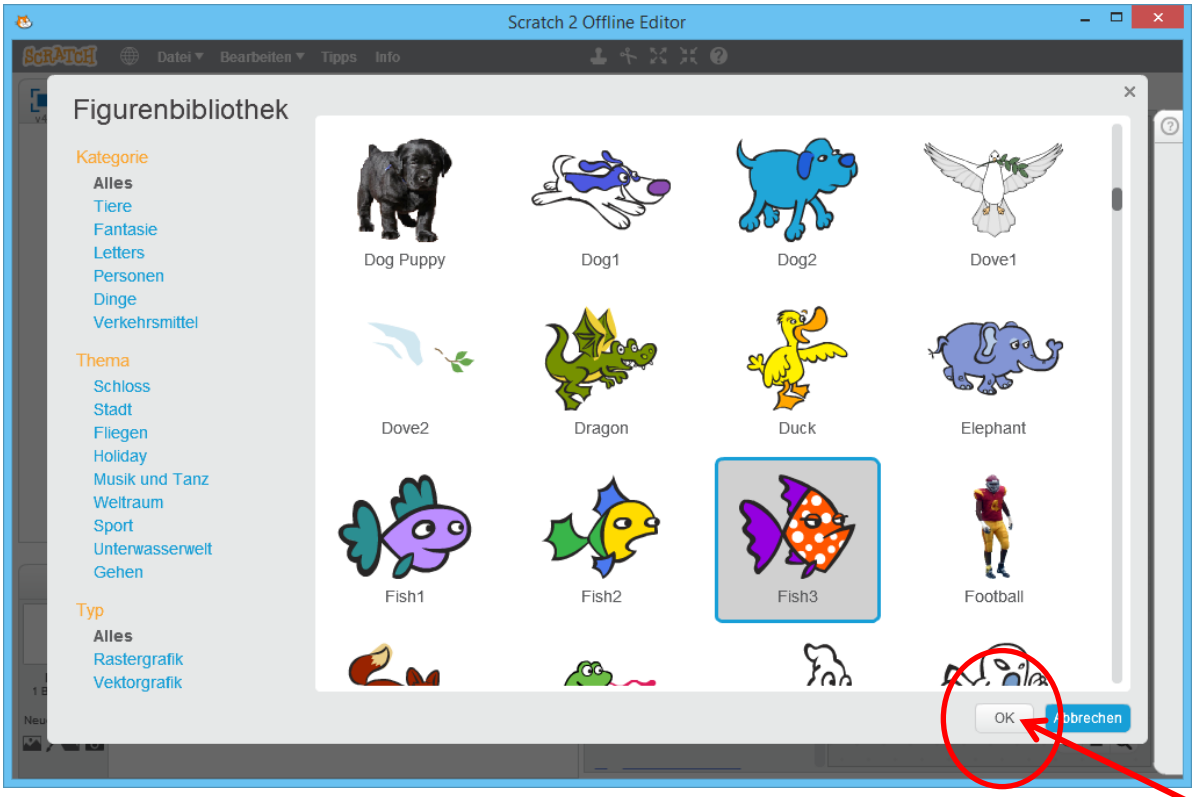
- Dazu klickt ihr mit der Maus einfach auf diese Schaltfläche:



CC BY SA

Es öffnet sich die **Figurenbibliothek.**

- Durchstöbert die Figurenbibliothek und wählt euch jetzt eine Figur durch Klick auf die linke Maustaste aus.

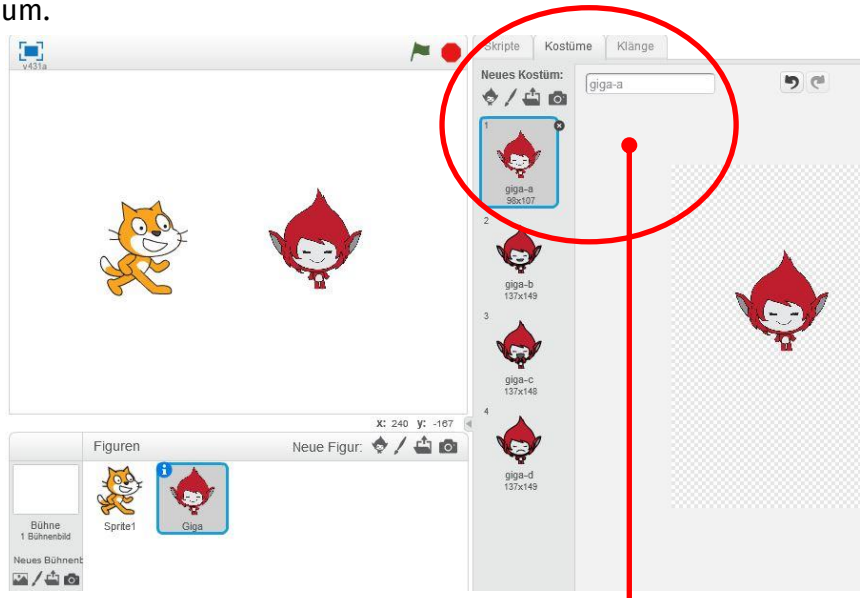


- Bestätigt eure Auswahl jetzt mit einem *Doppelklick*, oder klickt auf die Schaltfläche „OK“.

Gefällt euch eure Auswahl doch nicht könnt ihr sie hier über einen Rechtsklick, mit der Maus auf der Figur, löschen.

CC BY SA

Legt jetzt für eure Figuren einen passenden **Kostümnamen** an bzw. ändert den **vorhandenen Kostümnamen** (hier „giga-a“) um.



Achtung:
Dieses Symbol ist zweimal vorhanden!
Durch klicken auf dieses Symbol rechts unter eurer Bühne könnt ihr eine **neue Figur** anlegen.
Durch klicken auf das Symbol im Menü Kostüme, könnt ihr eurer Figur lediglich ein neues **Kostüm zuweisen.**

CC BY SA

Erzeugt einen Ton:

Jetzt wollen wir von unseren Figuren etwas hören! Dazu benötigen wir aus *Skripte* erst einmal einige Puzzleteile aus **Ereignisse** und **Klang** .

Damit eure Figur (hier die Katze) weiß wann es losgeht, legen wir eine **Startbedingung** fest.
Hierzu wählt ihr im Menü **Skripte**, aus den **Ereignissen** die Startbedingung „Wenn-Taste-Leertaste-gedrückt“ und zieht sie einfach rechts daneben, in eure **Programmieroberfläche**.

- Lasst die aus **Klang** ihr „*Miau*“ abspielen (habt ihr eine andere Figur, seht ihr auf der nächsten Seite, wie man ein passendes Geräusch auswählt).
- Testet euer Programm indem ihr auf die **Leertaste** drückt…
- Jetzt lasst die Katze **ein Instrument** ausprobieren.
- Testet euer Programm erneut ☺

CC BY SA

Scratch - Gruppenpuzzle -

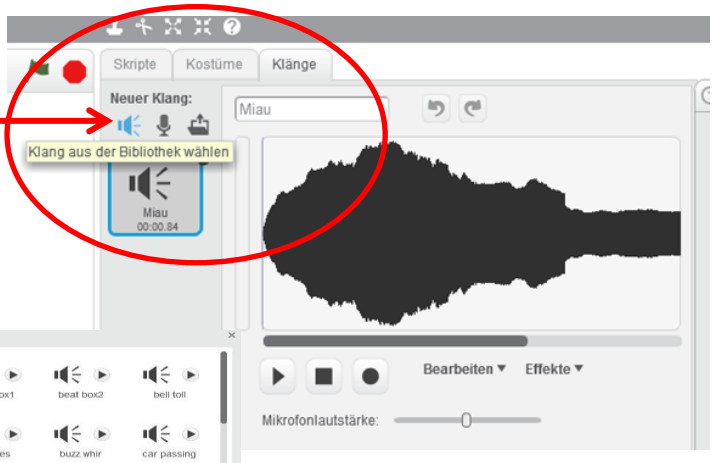
Klang und Kommunikation

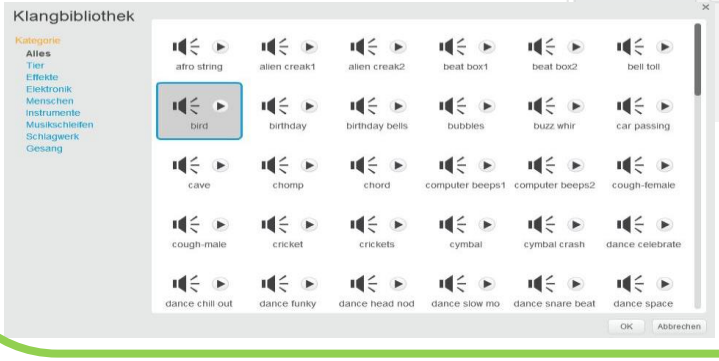
Was tut dieses Programm? Erklärt es mit euren eigenen Worten!

- Notiert eure Überlegungen hier:

- Versucht jetzt in **einem Programm** mehrere **Klänge** miteinander zu verknüpfen.

Sucht euch nun im Menü, unter **Klänge**, ein neues Geräusch aus der **Klangbibliothek** heraus. Klickt dazu auf den Lautsprecher.





CC BY SA

Klang und Kommunikation

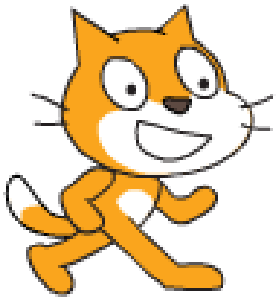
Im Menü **Skripte** unter **Klang** findet ihr euer ausgesuchtes Geräusch wieder.

Verwendet es jetzt in einem neuen Programm!

Aufgepasst! Der neue Klang gilt nur für die Figur mit der ihr zuletzt gearbeitet habt!

Wollt ihr einen neuen Klang für eine **andere Figur** benutzen, müsst ihr diesen erst **neu anlegen**!
Dazu klickt ihr vorher einfach auf die Figur mit der ihr Klänge erzeugen wollt und los geht's!

Klickt auf mich, um mir einen neuen Klang zuzuordnen!



Gut gemacht! Kurz verschnaufen und auf zum Endspurt!!

CC BY SA

Nachrichten versenden:

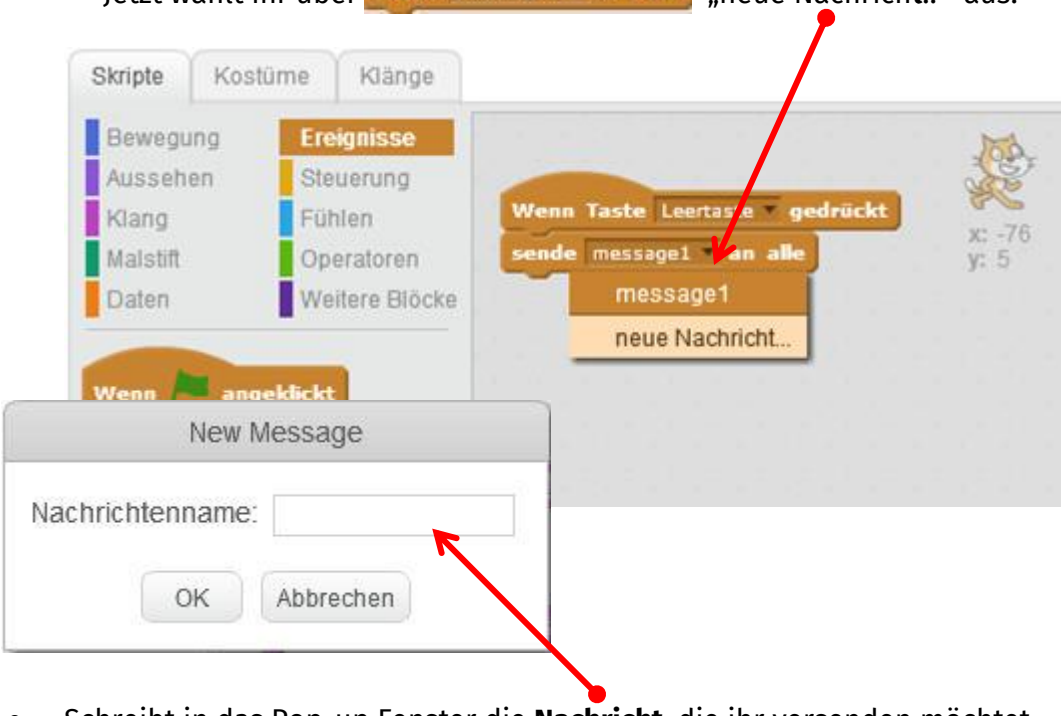
Nun müsst ihr euch entscheiden welche eurer Figuren eine Nachricht **versenden** und wer sie **empfangen** soll. Das Skript des Senders wird zuerst geschrieben.

Im Beispiel beginnen wir mit der , sie ist also unser **Sender**.

Wir benötigen Puzzleteile aus **Ereignisse**, **Steuerung** und **Klang**.

- Zieht aus **Ereignisse** die Puzzleteile „*Wenn-Taste-Leertaste-gedrückt*“ und „*sende-message1-an-alle*“ in eure Programmieroberfläche.
- Jetzt wählt ihr über

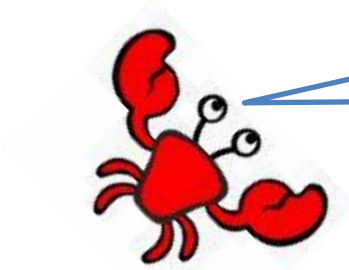
„neue Nachricht.. “ aus.



- Schreibt in das Pop-up Fenster die **Nachricht**, die ihr versenden möchtet.
- Wenn ihr jetzt die Leertaste an eurem PC drückt, wird eure Nachricht versendet.

Was müsst ihr tun, damit der Empfänger sich jetzt auch angesprochen fühlt?

CC BY SA



Genau! Ihr müsst mir als Empfänger ein Programm schreiben, damit ich weiß, dass ich etwas empfangen soll.

- Sucht in **Ereignisse** die Startbedingung für den **Empfänger** mit eurer Nachricht aus,

- Hängt aus **Steuerung** ein „*warte-1-Sek.*“ an die Startbedingung.
- Wählt zuletzt ein Puzzleteil aus **Klang**.
- Testet eure Programme!

Achtet darauf bei mehreren, parallelen Programmen des **Senders** jeweils die **gleiche Startbedingung** zu nutzen.
(Im Beispiel die Leertaste)

Alle Programme – von Sender und Empfänger – lassen sich dadurch **gleichzeitig** mit einem Klick starten.

Achtet darauf, dass ihr die Figur, für die ihr ein Programm schreiben möchtet, zuvor auch markiert habt!

CC BY SA

Wenn eure Programme so oder so ähnlich aussehen, dann habt ihr es geschafft!

Zusatzaufgabe (für alle die noch Puste haben):

Versucht eure Figur nach Empfang der Nachricht mehrere **Klänge** spielen zu lassen. Ihr könnt auch noch eine/mehrere weitere Figur/en erstellen und diese ebenfalls auf eure gesendete Nachricht mit einem Klang reagieren lassen; dann habt ihr so zu sagen ein ganzes Orchester ☺

Herzlichen Glückwunsch!
Ihr habt euer Ziel erreicht!

CC BY SA

Scratch - Das Spiel -

„Fangen“

„Fangen“:

Jetzt geht’s darum als **Expertengruppe** gemeinsam ein **Fang-Spiel** zu programmieren.
Dafür muss jeder von euch sein zuvor erlerntes *Expertenwissen* einbringen!
In dem Spiel soll **eine Figur** eurer Wahl **eine andere Figur** eurer Wahl über die Bühne jagen und fangen.
Die Folgenden Stichpunkte sollen euch bei der Umsetzung dieser Aufgabe helfen. Eurer Kreativität dürft ihr natürlich freien Lauf lassen.

Die vier Experten:

Um besser abschätzen zu können welcher Experte an welchem Punkt besonders sein Wissen einsetzen kann, werden im Verlauf des Arbeitsblatts die jeweiligen Experten explizit angesprochen.

- Experte **BZ**: Bewegung und Zeichnen
- Experte **KK**: Klang und Kommunikation
- Experte **FV**: Fühlen und Variablen
- Experte **GA**: Grafik und Aussehen

Grundstruktur des Spiels:

BZ GA KK

1. Erstellt eine Spielfigur, die später der **Fänger** sein soll (z.B. die
Passt die Größe der Figur eurem Hintergrund an und programmiert sie so, dass ihr sie mit den **Pfeiltasten** steuern könnt (**Ereignisse** und **Bewegung**).
Denkt daran, dass der Fänger **Auf-** und **Abtauchen** beherrschen muss.
Achtung: Verwendet

um die Laufrichtung des Fängers einzustellen.

GA

2. Erstellt eine weitere Spielfigur, die im Spiel gefangen werden soll (z.B. .
Passt die Figurengröße ebenfalls an euren Hintergrund an.

GA KK

3. Sucht euch **drei** verschiedene Bühnenbilder aus, die ihr in eurem Spiel verwenden möchtet (2 für den Spielverlauf, 1 für das Ende).

FV

4. Erstellt eine **Variable** für den Punktestand aus **Daten**.

FV

5. Fügt noch eine Stoppuhr aus **Fühlen** hinzu (

.
optional

CC BY SA

Spielstart:

- Das Spiel soll gestartet werden, wenn ihr auf das klickt (**Ereignisse**).

FV
- Stoppuhr und Punktestand müssen bei Spielstart zurückgesetzt werden. Verwendet hierfür folgende Puzzleteile aus **Fühlen** und **Daten** (denkt auch an das „wiederhole-fortlaufend-Puzzleteil“):

GA
- Die Bühne soll das erste Bühnenbild anzeigen (**Aussehen**).

BZ
- Der Fänger soll bei Spielstart immer an der gleichen Position (x-/y-Koordinate) sein. Hierfür bietet sich z.B. die Mitte der Bühne, also Position **x=0** und **y=0** (**Bewegung**).

BZ
- Überlegt euch auch eine feste Anfangsposition für die zu fangende Figur (**Bewegung**).

KK
- Bei Spielstart soll ein schöner Ton erklingen (**Klang**).

Fangen:

Wenn der Fänger die andere Spielfigur **gefangen** (bzw. berührt) **hat**, soll Folgendes geschehen:

BZ
- Die gefangene Figur soll abtauchen und an einem **zufälligen Ort** auf der Bühne wieder auftauchen (**Bewegung**).

  Verwendet:

  um eine **Zufallszahl** zu erzeugen (**Operatoren**).

KK
- Die gefangene Figur soll an alle die Nachricht **versenden**, dass sie gefangen wurde (**Ereignisse**).

  Verwendet:

  auch für die **3** folgenden Punkte:

KK
- ➢ Es soll ein **Ton** abgespielt werden (**Klang**).

FV
- ➢ Der **Punktestand** soll **zeitgleich um 1 erhöht** werden (**Daten**).

GA
- ➢ Nachdem die Figur gefangen wurde, soll das Bühnenbild wechseln (**Aussehen**). **Wählt** jetzt **nur 2** eurer 3 Bühnenbilder aus!

CC BY SA

Spielende:

- Das Spiel soll nach 30 Sekunden enden, also sobald die **Stoppuhr mehr als 30** Sekunden anzeigt. **Verwendet hierfür**:

Bonus:

BZ
FV
GA

- Die zu fangende Figur soll zurück zu ihrer Ausgangsposition (**Bewegung**). Dann soll sie euch nach eurem Namen fragen (**Fühlen**). Außerdem teilt sie euch mit, wie oft ihr sie gefangen habt (**Aussehen**). Ihr benötigt dazu diese 5 Puzzleteile.

  Kombiniert sie in der richtigen Reihenfolge!

KK

- Wieder soll die zu fangende Figur eine Nachricht versenden um das Ende des Spiels zu verkünden (**Ereignisse**). **Verwendet wieder folgendes Puzzleteil:**

KK

- Das Spielende wird ebenfalls durch einen schönen **Ton** eingeläutet (**Klang**).

GA

- Die Bühne soll zum Abschluss des Spiels ein feierliches Bühnenbild (euer drittes) anzeigen (**Aussehen**).

CC BY SA

Herzlichen Glückwunsch!
Ihr habt euer Ziel erreicht!