

Scratch Kurs Design

1. Stunde: Einführung

1. Begrüßung

- *Hallo zusammen. Schön, dass ihr alle da seid.*
@Beobachter: Teilnehmerliste checken
Team vorstellen: Leitung sowie Betreuung
- Ablauf erklären zu Stunde und Kurs:
*Wir zeigen euch ein kleines **Beispiel für ein Programm in Scratch**, damit ihr ungefähr wisst, was wir hier in dem Kurs lernen wollen.*
*Es geht dann weiter mit den **Grundlagen von Scratch** und einer **Aufgabe**, die wir heute zusammen erarbeiten möchten.*
Wir wollen dabei alle zusammen mitmachen, d.h. ihr könnt euch dann wie in der Schule auch melden und versuchen die Aufgaben zu lösen.
*Zum Schluss machen wir noch eine kleine **Umfrage** und wir möchten euch einen **Ausblick** auf die Stunde von nächster Woche geben.*
- *Falls ihr Fragen während der Stunde habt, ist das gar kein Problem: Beim Programmieren gibt es nicht nur die eine Lösung, sondern oft mehrere Wege und Fehler gehören dazu. Meldet euch einfach und dann kümmern wir uns darum.*

2. Intro: Programmierung und Beispiel (Spiel)

- *Frage in die Runde: Was ist eigentlich Programmierung? Was stellt ihr euch darunter vor? Vielleicht möchte jemand von euch kurz etwas dazu sagen oder seine/ihre Erfahrungen schildern?*
(Kurz warten, ob sich jemand meldet: Ggf. dynamisch reagieren, falls zu wenig oder zu viele Meldungen)
- *Programmierung kann total vielseitig sein: Viele verschiedene Anwendungsfälle sind möglich und es ist eine super kreative Tätigkeit. Zum Beispiel auch das Kreieren von Spielen und Animationen ist möglich, worauf wir uns hier im Kurs konzentrieren möchten. Andere Bereiche, wofür Programmierung verwendet wird: Roboter, Apps auf dem Handy, viele andere Dinge, die ihr aus eurem Alltag kennt.*
- *In den nächsten 6 Wochen wollen wir zusammen die Grundlagen vom Programmieren mit Scratch lernen. Unser Ziel: kleine Spiele programmieren, wie zum Beispiel folgendes:*

- **Flappy Bird zeigen** und kurz anspielen: *Vielleicht kennt der eine oder die andere das Spiel: Ziel ist es, den Vogel mit Hilfe der Tastatureingaben durch die Hindernisse zu steuern ohne anzustoßen und dafür gibt es Punkte. Solche Spiele oder Geschichten könnt ihr am Ende unseres Kurses dann auch selbst programmieren! (yay). Ihr bekommt dann auch ein Zertifikat für den Kurs :) (again yay).*

Dann legen wir gleich mal los!

Zum Ablauf: Wir zeigen euch an unserem Bildschirm unsere Anleitungen und danach geben wir euch Zeit, dann könnt ihr es selbst machen.

3. Intro: Scratch anhand Scratch-Katze

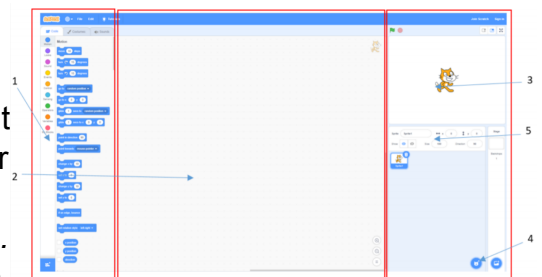
- (Unser Bildschirm ist immer noch geteilt): *Wir zeigen auch jetzt kurz wie man sich einloggt und danach könnt ihr es gleich selbst ausprobieren.*
- *Geht in euren Browser und gebt die Adresse <https://scratch.mit.edu/> ein. Das sollten sie hoffentlich schon alle gemacht haben (laut meiner Mail)*
- *Ihr habt von uns bereits einen Benutzernamen bekommen und mit dem loggt ihr euch jetzt ein (Wir loggen uns auch mit unserem ein).*
- **Rückfrage: Habt ihr alle euren Benutzernamen und seid ihr eingeloggt?** (Annahme: Nicht alle haben das bereits im Voraus geschafft und es vergeht einige Zeit *“Wir warten kurz, bis sich alle eingeloggt haben”*). Zur Erinnerung: Die Kinder müssen sich bei der ersten Anmeldung mit PW “InfoDidUPTeacher” anmelden und danach ein eigenes vergeben. Bei Vergessen können wir das über den Teacher-Account zurücksetzen, dann ist das PW “InfoDidUPTeacher”.

Screensharing AN

- Klickt auf Erstellen“, um euer erstes Projekt zu erstellen: Gebt gleich einen Titel ein, z.B. “Mein erstes Programm” und drückt daneben auf den Button “Veröffentlichen”. Somit ist euer Projekt auch für alle anderen sichtbar. Dann könnt ihr mit dem Button “Schaut hinein” wieder zur Bearbeitung wechseln. **WICHTIG: Bitte speichert regelmäßig euer Projekt ab. Geht dazu in der Leiste auf “Datei speichern”**
- Wir fangen als nächstes an, die Scratch-Oberfläche zu erklären

Links: Blockübersicht

- Blöcke, aus denen ein Programm gebaut werden kann, werden in diesem Reiter angezeigt.
- Es gibt *unterschiedliche Kategorien* wie z.B. *Bewegung, Fühlen, Steuerung und Ereignisse*.



- Diese *Blöcke können in die Mitte gezogen werden*, wo die Bearbeitungsfläche ist.
- Mit Hilfe der Blöcke können wir zum Beispiel die Figur, die wir rechts sehen, Aktionen ausführen lassen
- Die Schnittstellen/Steckplätze der Blöcke geben dabei die Verbindungsmöglichkeit an

→ ziehe den Block "Gehe 10er Schritt" in die Mitte, ändere die Schrittgröße auf 100

Hier können wir sehen, wie sich die Katze bewegt, wenn wir auf den Block klicken.

→ ziehe den Block "Sage 'Hallo' für 2 Sekunden" in die Mitte unter den anderen Block

Die Figur kann durch Sprechblasen etwas sagen.

→ setze nun noch den Block "Wenn grüne Flagge angeklickt wird" vor den ersten Block

Wenn wir diesen Block verwenden, können wir das Programm auch über die grüne Flagge rechts oben starten.

Info: Die aneinandergereihten Anweisungen nennt man auch Skript.

- Screensharing AUS: Kiddies sollen das oben selbst machen

Bitte versucht nun auch ihr, die gleichen Blöcke in die Mitte zu ziehen und die Schrittgröße zu verändern.

Rückfrage: Hat das bei allen geklappt? JA?

Screensharing AN

Rechts oben: Darstellungsbereich / Bühne

Es können verschiedene Bühnenbilder ausgewählt werden.

Rechts unten: Figur (auch Sprite genannt)

Verschiedene Figuren können über den blauen Button ausgewählt werden.

Wenn man die Figur ausgewählt hat, sieht man rechts überhalb der Figur den Papierkorb. Klickt man hier drauf, wird die Figur gelöscht. → zeigen

Das blaue Kästchen links unten: Erweiterungen wie zB Malen oder Sprache möglich.

4. Aufgabe 1: Zauberer

Nun kommen wir zu dem Programm, mit dem wir uns heute beschäftigen möchten:

THEMA DER STUNDE: DER ZAUBERER

Ziel der heutigen Stunde: Den Zauberer sprechen und verschwinden lassen. Wenn wir noch Zeit haben, dann lassen wir Zauberer sogar noch in einen Hasen verwandeln. Kurze Anmerkung: In dieser Stunde werden wir euch noch sehr stark anleiten, damit ihr einen ersten Eindruck von Scratch bekommt und in den anschließenden Stunden dürft ihr dann immer selbstständiger und total kreativ arbeiten!

Also, heute sollte euer Programm so ausschauen: Kurz das Endergebnis zeigen.

Aufgabe 1.1:

Also fangen wir an: Erstellt wieder ein neues Projekt (Datei -> Neu). Gebt dem Programm einen Titel wie z.b. "Der Zauberer" oder "Der Zauberwald" und drückt auf den Button "Veröffentlichen".

*Als erstes **löscht ihr bitte die Katze**. Wie das geht, habe ich euch vorher erklärt.*

Erstellt selbst euer erstes Bühnenbild. Versucht, mein Beispiel nachzumachen. Zunächst sollt ihr nur den Bühnenhintergrund und die Figur auf den Bildschirm bringen.

Screensharing AUS



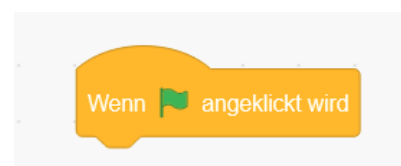
Bühnenbild: forest
Figur: wizard

Probiert das selbst.. Hat das jeder geschafft?

Aufgabe 1.2:

Screensharing AN

Als Beginn verwenden wir einen "wenn grüne Flagge geklickt" Block, damit wir das Programm einfach starten können.



Als nächstes müssen wir den Zauberer nach links setzen, von wo aus er dann auf die Bühne gleitet: Zauberer mit der Maus ganz nach links an den Rand ziehen.

Damit der Zauberer von Anfang an immer auf diese Stelle gesetzt wird, müssen wir das über unser Programm an den Computer kommunizieren.

Wir verwenden dazu den Block “gehe zu x: ... y: ...” aus der Kategorie ‘Bewegung’. Wenn euer Zauberer gerade schon ganz links steht, könnt ihr einfach den blauen Block von links in die Mitte ziehen, die richtigen Koordinaten sollten dann schon eingetragen sein.

Koordinaten:

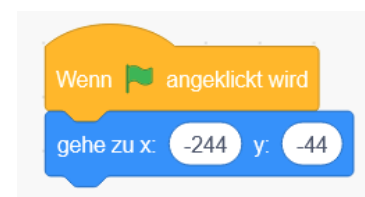
x-Position zeigt an, wie weit links oder rechts unsere Figur ist

y-Position zeigt, wie weit unten oder oben

*Bitte versucht, den **Zauberer jetzt soweit wie möglich nach links zu setzen.***

Bitte fügt nun den “gehe zu” Block in euer Programm ein. Bitte schaut nach vorne, wenn ihr soweit seid.

Screensharing AUS



Screensharing AN

Im nächsten Schritt soll der Zauberer in die Mitte der Bühne gleiten.

Frage: Hat jemand einen Vorschlag, wie wir das machen könnten?

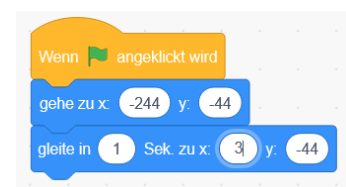
Tipp (falls niemand drauf kommt): schaut euch mal die Blöcke in der Kategorie “Bewegung” an.

Wir brauchen nun den Block “**gleite in ... Sekunden zu x y**”. Als x-Position verwenden wir 3 und für die y-Position -44, damit der Zauberer in der Mitte ankommt. Ihr könnt auch andere Werte für die Positionen verwenden, schaut einfach, dass sich der Zauberer in der Mitte befindet.

Bitte fügt diesen Block nun mit den richtigen Werten hinzu und lasst mich dann über die Reaktionen wissen, wenn ihr damit fertig seid.

Wenn es Fragen gibt, meldet euch gerne zu Wort oder schreibt in den Chat.

Wenn alle soweit sind, Erinnerung an **Speichern des Projekts**.



Screensharing AUS

Aufgabe 1.3:

Screensharing AN

Als nächstes soll der Zauberer sagen "Hallo, ich zeige euch jetzt einen Trick!".

Frage: Welchen Block könnten wir hierfür verwenden?

Es wird der **"sage ..."-Block** verwendet. Indem man auf das weiße Feld im Block klickt, kann man die Aussage der Figur verändern.

Nun soll dem Zauberer eine Stimme gegeben werden, damit man ihn auch sprechen hören kann: **Erweiterung „Text zu Sprache“** zeigen.

Frage: Welche Blöcke in der "Text zu Sprache"- Kategorie sind eurer Meinung nach sinnvoll hier einzufügen? Die Stimme soll zum Zauberer passen und man soll das hören können, was er auch oben in der Sprechblase sagt.

Bitte fügt diesen Block hinzu und gebt mir eine Reaktion, wenn ihr fertig seid.

-> "ändere die Stimme zu ..." (Riese oder Tenor)

-> sage "Hallo, ich zeige euch jetzt einen Trick!"

Screensharing AUS

Falls es noch Fragen gibt, scheut euch nicht, diese zu stellen!

Bitte speichert das Projekt nochmal ab.

Screensharing AN



Vielleicht ist dem einen oder anderen aufgefallen, dass der Zauberer schon fast das sprechen beginnt, während er noch in die Mitte gleitet.

Frage: Was könnten wir noch für einen Block in unser Programm einfügen, damit ein bisschen Zeit zwischen der Bewegung und dem Sprechen vergeht?

Dafür muss der **"warte 1 Sekunde"-Block** aus der Kategorie "Steuerung" **vor dem lila Block "sage ..."** eingesetzt werden.

(Wir verwenden nicht den “sage für 1 Sek”-Block, weil die Sprechblase und der gesprochene Text sonst nicht parallel ablaufen)

*Bevor der Zauberer weiterspricht, soll er auch wieder eine kurze Pause machen.
Bitte vervollständigt euer Programm und sagt mir dann, wie ihr das gelöst habt.*

-> “warte 1 Sekunde” **auch nach den zwei grünen Sprache-Blöcken** einfügen

Wir möchten nun, dass die Sprechblase wieder verschwindet, nachdem der Zauberer gesprochen hat.

Frage: Hat jemand eine Idee, wie man die Sprechblase wieder verschwinden lassen kann?

-> leeren “Sage ...”-Block einfügen

Screensharing AUS

Ist jeder soweit?

Screensharing AN

Aufgabe 1.4:

Der Zauberer soll nun „Abrakadabra“ sagen (mit Sprechblase und Stimme).

*Wir haben ja jetzt schon gelernt, wie der Zauberer sprechen kann. Bitte versucht nun eigenständig, den Zauberer “Abrakadabra” mit Sprechblase und Stimme sagen zu lassen.
Hinweis: Die Stimme ist nun schon in der “Text zu Sprache”-Erweiterung eingestellt und muss nicht noch einmal extra gesetzt werden.*

Screensharing AUS

Kurz Zeit lassen und dann die Lösung zeigen:

Screensharing AN

-> Lila “Sage ...”-Block einfügen

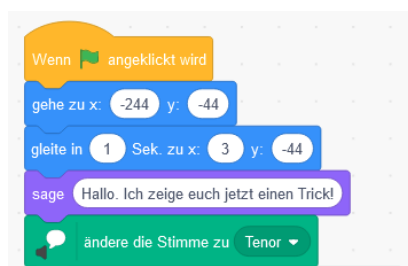
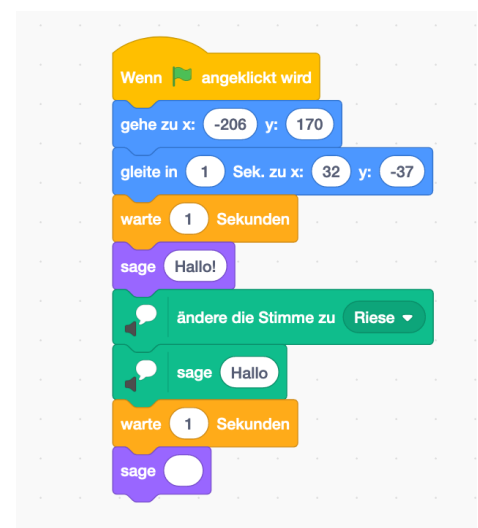
-> Grünen “Sage ...”-Block einfügen

-> “Warte 1 Sekunde” einfügen (damit die Sprechblase nicht direkt wieder verschwindet durch den folgenden Block)

-> Lila “Sage”-Block ohne Inhalt einfügen

*Hat diese Aufgabe jede*r geschafft? Gibt es Fragen?*

Wenn alle fertig sind, bitte nochmal das Projekt speichern.



Nun soll der Zauberer von der Bühne verschwinden.

*Frage: Wie könnte das funktionieren? (Wir haben Kategorie Aussehen offen):
Welcher Block könnte denn hier passen?*

-> **“Verstecke dich”-Block hinzufügen**

Projekt nochmal ausführen, damit man sieht, dass der Zauberer nicht mehr erscheint, wenn man es wieder startet.

Damit der Zauberer nicht für immer versteckt bleibt, nachdem wir unser Programm einmal abgespielt haben, müssen wir auch eine Anweisung einfügen, die bewirkt, dass der Zauberer am Anfang immer erscheint.

An welcher Stelle könnten wir welchen Block einfügen?

-> **“zeige dich”-Block ganz an den Anfang des Skriptes setzen**

Außerdem wollen wir noch einen Klang hinzufügen zu dem Teil des Programms, an dem der Zauberer verschwindet. Der Klang “magic bell” ist hier passend.

Wer kann mir sagen, welchen Block wir dafür auswählen müssen?

-> **“Spiele Klang ‘magic bell’ ganz” hinzufügen**

Dann probiert das doch mal selbst aus.



Screensharing AUS

*Alle fertig? Dann ist der größte Teil schon einmal geschafft!
Toll gemacht!*

Bitte an dieser Stelle noch einmal das Projekt speichern.

Aufgabe 1.5: ZUSATZAUFGABE FÜR DAHEIM

(Wir schicken ihnen diese Erweiterung der Aufgabe als PDF in einer Mail zuschicken und dann können sie das gerne zu Hause

nochmal selbst ausprobieren.)

Screensharing AN: Programm zeigen

Wir möchten jetzt, dass der Zauberer sich in einen Hasen verwandelt.



Diesen Teil des Programms sollt ihr nun einmal versuchen, alleine zu schreiben.

Hinweise: Es muss für den Hasen eine neue Figur hinzugefügt werden. Diese Figur soll sich nach circa 10 Sekunden zeigen (ausprobieren). Wenn der Hase erschienen ist, soll er (mit Stimme und Sprechblase) sagen "Toll, nicht wahr?"

Screensharing AUS

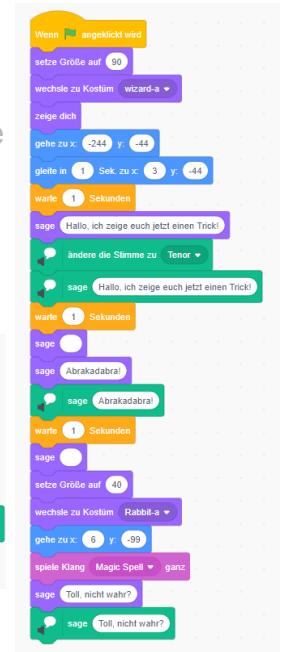
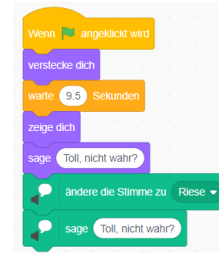
-> Hinweise in Chat schreiben

Wenn jemand schon schneller fertig ist mit dieser Aufgabe, darf er/sie auch noch weitere Zaubertricks eigenständig programmieren

- > **Neue Figur „rabbit“** hinzufügen
- > **Rechts dargestelltes Skript** hinzufügen
- > **Parallelität der Programme mit grüner Flagge** zeigen

Fügt bitte den gelben Block "Wenn grüne Flagge angeklickt wird" an den Anfang eurer beiden Anweisungsblöcke.

Wenn wir nun die grüne Flagge über der Bühne anklicken, laufen beide Programme gleichzeitig ab und wir haben unser Endergebnis.



Falls noch Zeit ist: Kiddies sollen ihre super duper Projekte zeigen.

5. Ende der Stunde: Umfrage

- Damit wir einschätzen können, wie schwer oder leicht euch die Aufgaben gefallen sind und ob ihr Spaß hattet, machen wir noch eine kleine Umfrage.

6. Ausblick

- Super, danke für euer Feedback! Wir hoffen, ihr hattet Spaß und habt schon etwas gelernt.
- In der nächsten Stunde fangen wir dann mit dem ersten richtigen Programmierkonzept an: Wiederholungen. Außerdem wird es tierisch am Nord- und Südpol, also seid gespannt!....
- Wir wünschen euch ein schönes Wochenende!

2. Stunde: Wiederholungen

1. Begrüßung

- *Hallo zusammen. Schön, dass ihr wieder da seid.*
@Beobachter: Teilnehmendenliste checken

- Kurze Retrospektive zu letzter Stunde? Fragen nach Hausaufgabe: *Habt ihr alle den Hasen noch herbeigezaubert? Gab es Probleme, die ihr kurz besprechen möchtet?*

2. Einführung heutiges Thema

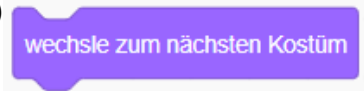
- Was erwartet euch heute? Wiederholungen (Kontrollstrukturen). Dafür zeige ich euch kurz zwei Programme.

Screensharing AN Beide Beispielprojekte zeigen

3. Aufgabe 1: Tanz

Aufgabe: 1.1:

Wir wollen die Figur tanzen/laufen lassen (je nach Kostümen)

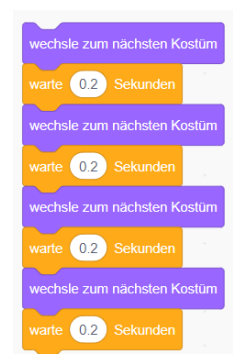
- Wir erstellen ein neues Projekt mit Titel “Tanz” (o.ä.) und wählen die Figur sowie die Bühne aus.

- Wir verwenden für die Änderungen der Haltung (‘tanzen’) den Block “Wechsle zum nächsten Kostüm” aus der Kategorie “Aussehen”. In Scratch werden die verschiedenen “Haltungen” / wie die Figur aussieht als Kostüm bezeichnet, d.h., wenn wir nicht möchten, dass die Figur sich von einer Stelle zu einer anderen Stelle bewegt, sondern nur ihre Haltung ändert, suchen wir nicht in der Kategorie “Bewegung”.

Die Figuren haben unterschiedlich viele Kostüme. Manche zum Beispiel haben nur ein oder zwei Kostüme, andere wiederum sogar acht!

Wir haben auch die Möglichkeit zu einem bestimmten Kostüm zu wechseln.

In unserem Beispiel jetzt möchten wir aber alle Kostüme einer Figur verwenden.

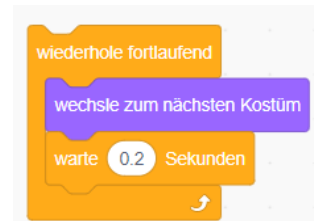
- Damit die Figur ihre Haltung fortlaufend verändert, müssen wir sie das Kostüm immer wieder wechseln lassen. Also setzen wir also einfach diese Anweisungen mehrere Male hintereinander.
- Damit wir die Bewegungen/Änderungen sehen können, müssen wir nach dem Kostümwechsel immer kurz warten.



Wir verwenden hier die Milchtüte, um den Kostümwechsel zu demonstrieren.

Das Programm würde dann ungefähr so aussehen. Und wenn wir die Figur sich länger bewegen lassen möchten, wird das ein ganz schön großer Aufwand und unübersichtlich.

- Es gibt beim Programmieren eine Lösung für dieses Problem: die Wiederholungen (auch Schleifen genannt).
 - Wir packen unsere Anweisung in einen Wiederholungsblock, der angibt, wie oft wir das Ganze wiederholen möchten. Das Skript von oben können wir also durch eine Schleife so ausdrücken, wie hier rechts zu sehen ist.
 - (Blöcke in Programm einfügen und laufen lassen)
-
- Es gibt verschiedene Arten von Wiederholungen: die endliche, bei der wir eine bestimmte Anzahl von Wiederholungen eingeben und die unendliche, die solange läuft, bis wir das Programm manuell stoppen
 - Wir wählen hier nun erst einmal die fortlaufende Schleife
 - Das schaut doch schon viel strukturierter aus und ist eigentlich ganz einfach oder?
 - Probiert das doch gleich mal selbst aus und sucht euch bei der Kategorie „Tanz“ eine beliebige Figur aus und versuche, sie mit einer Wiederholung tanzen zu lassen. Wählt außerdem ein passendes Bühnenbild aus der Kategorie „Musik“.



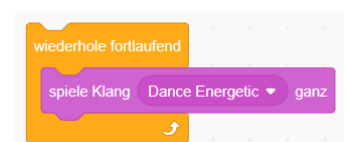
Screensharing AUS

Aufgabe 1.2:

- Als nächstes wollen wir, dass unsere Figur auch zu Musik tanzen kann.

Screensharing EIN

- Wir wählen eine neue Figur aus, in diesem Fall eine Musikbox (radio). Zu dieser fügen wir einen beliebigen Klang hinzu (bei Klängen „dance“ eingeben für passende).
- Hier: zeigen, wie man auch selbst Klänge aufnehmen kann
- Frage: Wie schaffen wir es, dass unsere Musik durchgehend läuft, solange unsere Figur tanzt?
- Richtig, wir verwenden eine fortlaufende Wiederholung dafür.



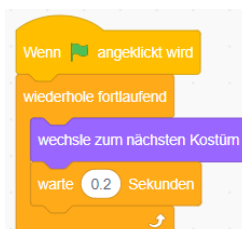
- Außerdem hat auch die Musikbox zwei verschiedene Kostüme. Bitte fügt ebenfalls ein Skript hinzu, damit die Kostüme sich fortlaufend ändern. Ihr wisst jetzt schon wie das geht!

Screensharing AUS

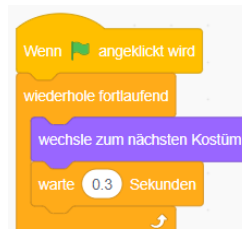
- Frage: Wer erinnert sich noch, was wir tun können, damit alle Skripte gleichzeitig starten?

Screensharing AN

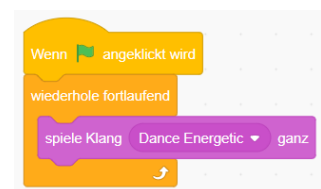
-> grüne Flagge vor jedes Skript



Tänzer*in



Musikbox Kostüm



Musikbox Klang

Aufgabe 2: Eisbär und Pinguin

“Neu” anklicken damit wir ein neues Programm anlegen

Um noch ein bisschen weiter die Wiederholungen zu üben, wollen wir als nächstes eine Treppe bauen, damit der Eisbär den Pinguin besuchen kann.

Das Endergebnis soll wie folgt aussehen: (kurz Ergebnis in Scratch zeigen)

Screensharing AUS

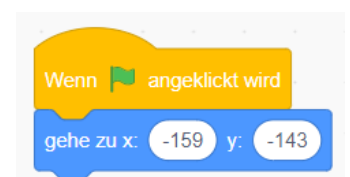
Aufgabe 2.1

Als erstes brauchen wir unsere Figuren. Bitte erstellt das Bühnenbild (slopes) und fügt dann den Eisbär und den Pinguin hinzu. Ändert die Größe der Figuren jeweils auf 60. Setzt den Eisbären nach links unten und den Pinguin rechts oben auf den Berg.

Screensharing AN

Frage: Wer erinnert sich noch, was wir tun müssen, damit der Eisbär beim Starten des Programms immer an dieselbe Stelle zurückgeht?

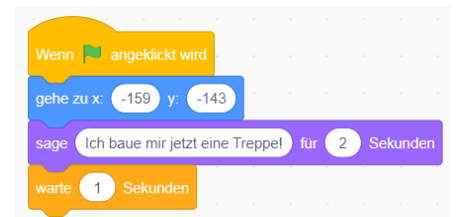
-> wir fügen einen “gehe zu”-Block ein



Screensharing AUS

Seid ihr alle soweit?

Außerdem möchten wir, dass der Eisbär durch eine Sprechblase sagt "Ich baue mir jetzt eine Treppe!" und danach eine Sekunde wartet, bevor es damit losgeht.



Screensharing AN

Aufgabe 2.2

Als nächstes möchte ich euch eine neue Erweiterung zeigen. Letztes Mal haben wir schon die Erweiterung "Text zu Sprache" verwendet.

Klickt wieder links unten auf den blauen Button für die Erweiterungen und wählt "Malstift" aus. Diese Erweiterung werden wir verwenden, um die Treppe zu erstellen.

Als erstes müssen wir den Stift einschalten, um ihn verwenden zu können. Dann wollen wir die Farbe auf dunkelblau und die Stiftdicke auf 4 einstellen.

Frage: Welche Blöcke könnten wir dafür verwenden?



Screensharing AUS

Bitte fügt nun auch die besprochenen Blöcke zu eurem Programm hinzu.

Screensharing AN

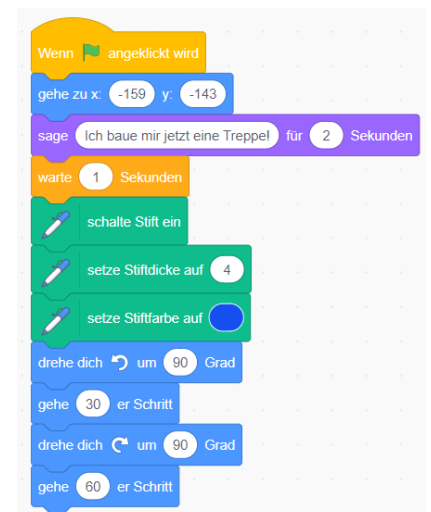
Aufgabe 2.3

Frage: Hat jemand eine Idee, wie ich nun eine Stufe malen könnte?

Die Figur muss sich um 90 Grad nach links drehen und dann 30 Schritte gehen, um den Strich nach oben zu malen.

Dann muss sie sich wieder um 90 Grad nach rechts drehen und 60 Schritte gehen, um den Strich nach rechts zu malen.

Da wir vorher den Stift eingeschaltet haben, erhalten wir somit eine Stufe.



Bitte fügt alle diese vier Blöcke hinzu.

Aufgabe 2.4

Frage: Was müssen wir nun tun, damit fünf Stufen gemalt werden?

-> Wir brauchen eine Schleife / einen Wiederholungs-Block.
Welchen?

Ohne den Wiederholungs-Block müssten wir die blauen Anweisungen (um die Stufe hinzuzufügen) fünf mal hintereinander schreiben. Wir sparen uns also einiges an Arbeit, wenn wir die Wiederholung verwenden.

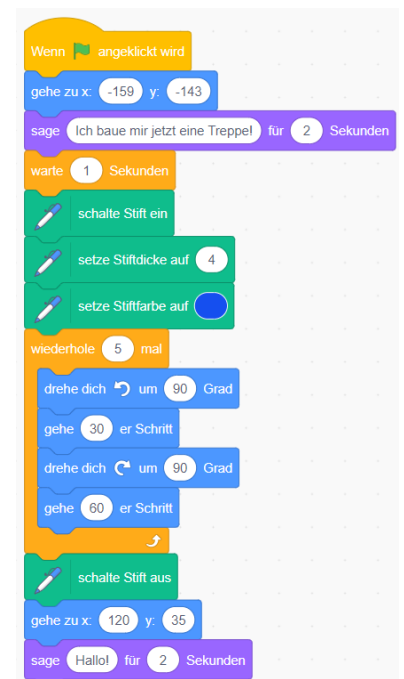
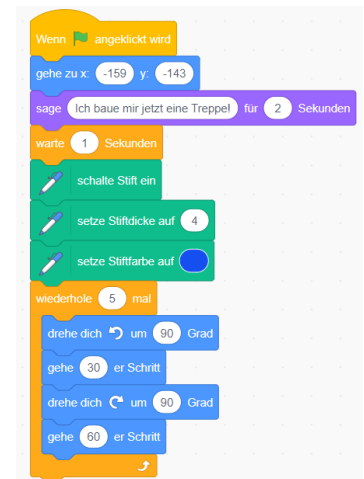
Bitte fügt nun auch die Schleife zu eurem Programm hinzu.

Screensharing AUS

Screensharing AN

Wir müssen nun den Stift wieder ausschalten, da der "Mal-Prozess" abgeschlossen ist.

- Voraussichtliches Ende der Stunde; falls noch mehr als 10 Minuten Zeit sind, könnt ihr Aufgabe 2.5 machen bevor ihr zur Umfrage kommt -



Aufgabe 2.5

Außerdem möchten wir, dass sich der Eisbär auf die obere Kante der letzten Stufe stellt und nicht darunter stehen bleibt.

Daraufhin soll er zum Pinguin "Hallo!" sagen.

Frage: Welche Blöcke brauchen wir dafür?

Füge “schalte Stift aus” hinzu.

Füge “gehe zu...” hinzu (um die richtige Position zu finden, einfach den Eisbären mit der Maus an die richtige Stelle ziehen).

Füge “sage ... für 2 Sek”-Block hinzu.

Screensharing AUS

Bitte fügt die entsprechenden Blöcke nun hinzu.



Screensharing AN

Frage: An welcher Stelle könnten wir noch eine Anweisung hinzufügen, damit bei jedem Programmstart die Treppe wieder verschwindet? Welche Anweisung wäre hier geeignet?

Wir fügen den Block “Lösche alles” aus der Kategorie “Malen” direkt am Anfang des Skripts hinzu.

Außerdem möchten wir, dass der Eisbär sich ein bisschen mehr Zeit lässt beim malen der Stufen.

Frage: Was wäre dafür eine geeignete Anweisung?

-> “warte 0.3 Sekunden” nach jeder Stufe

Screensharing AUS

Bitte fügt nun auch diese beiden Blöcke hinzu.



Wir haben uns heute ja auch schon mit dem Kostümwechsel beschäftigt. Auch der Eisbär hat verschiedene Kostüme und kann Laufbewegungen darstellen.

Frage: Welchen Block brauchen wir dafür und wo fügen wir ihn ein?

-> “wechsle zum nächsten Kostüm”, innerhalb des Wiederholungsblocks, damit bei jeder Stufe ein Schritt dargestellt wird

Außerdem fügen wir direkt nach der Schleife ein “wechsle zu Kostüm polar-bear a” ein, damit der Eisbär auf der letzten Stufe wieder in seiner Ausgangsposition steht.

Screensharing AUS

Fügt bitte diese beiden Blöcke hinzu.



Screensharing AN

Das Programm sollte jetzt insgesamt so aussehen:

Damit haben wir den wichtigsten Teil geschafft.

Erweiterungsaufgabe für DAHEIM

Wir möchten wir nun, dass der Bär nach jeder Stufe ein Salto macht, sich einmal um sich selbst dreht.

Außerdem wollen wir auch hier bei jeder Stufe die Laufbewegungen vom Eisbären sehen. Welche Blöcke könnten wir dafür einbauen?

Frage: Was für eine Anweisung brauchen wir, damit sich der Eisbär einmal um sich selbst dreht?

-> viermal "drehe dich um 90 Grad"

Natürlich würde auch eine 360-Grad-Drehung Sinn ergeben, allerdings würden wir die Drehung dann nicht sehen. Wenn wir uns also vier Mal um 90 drehen und dazwischen immer ganz kurz warten (0.1 Sek), sehen wir die Drehung auch.

Das heißt, wir möchten diese Drehung viermal wiederholen.

Frage: Welchen Block verwenden wir, um etwas viermal zu wiederholen?

Da wir die Drehung immer nach jeder Stufe einfügen möchten, befindet sich der Anweisungsblock innerhalb des anderen Wiederholungsblocks. Wir haben also eine Schleife in einer anderen Schleife. Dies nennt man dann "verschachtelte Schleifen".

Für die Laufbewegungen fügen wir bei jeder Stufen-Wiederholung ein "wechsle zum nächsten Kostüm ein".

Screensharing AUS

Bitte fügt nun auch die "wiederhole 4 mal", "warte 0.1 Sekunden" und "drehe dich um 90 Grad"-Blöcke in euer Skript ein.

Damit sind wir am Ende der Stunde angelangt. Schön, dass ihr wieder alle so toll mitgemacht habt!

Als "Entdeckungsaufgabe" für daheim: "lösche alles" und als Zusatzaufgabe Salto des Eisbären wenn er beim Pinguin ist und sich freut (auch dass sie sich begrüßen)

In der nächsten Stunde werden wir ein weiteres Konzept der Programmierung kennenlernen: Bedingungen. DA bleibt es tierisch und wir lernen FLIEGEN :)

3. Stunde: Bedingte Anweisungen

1. Begrüßung

- *Hallo zusammen. Schön, dass ihr wieder da seid.* (Ggf. kurz warten, bis Teilnehmendenanzahl erreicht ist.)
@Beobachter: Teilnehmendenliste checken
- **Kurze** Retrospektive zu letzter Stunde? Fragen nach Hausaufgabe: *Konntet ihr die Treppe malen lassen? Hat jemand auch das mit dem Salto geschafft?*

2. Einführung heutiges Thema

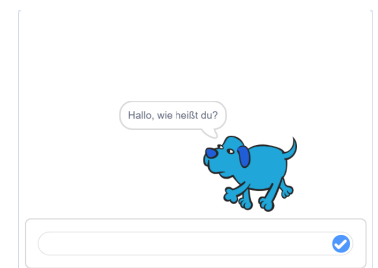
- Was erwartet euch heute? Bedingte Anweisungen.
Dafür zeige ich euch kurz ein kleines Spiel, das wir später erstellen werden.

Screensharing AN Spiel Flying Cat zeigen

3. Aufgabe 1: Hund Konversation

- Als erstes möchten wir aber mit einem anderen kleinen Programm anfangen, über das wir selbst mit einer Figur kommunizieren können.
- Ich zeige euch erst einmal, wie das am Ende aussehen soll.

-> Ablauf des Programms herzeigen (name eintippen)



Aufgabe 1.1:

Heute möchten wir eine neue Kategorie kennenlernen: Fühlen.

Wir möchten unsere Figur eine Frage stellen lassen.

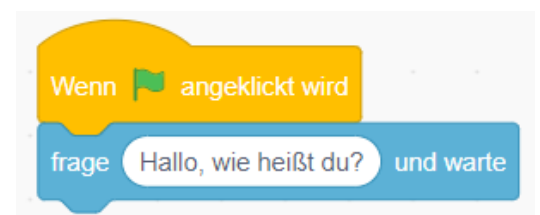
Frage: Welchen Block könnten wir dafür aus der Kategorie Fühlen verwenden?

-> Block "frage ... und warte" auswählen.

Wir fügen die Frage "Hallo, wie heißt du?" ein.

Screensharing AUS

Bitte fügt den Hund und diesen Block ebenfalls in euer Programm ein.



Aufgabe 1.2:

Screensharing AN

Da wir den frage-Block eingefügt haben, haben wir nun die Möglichkeit, in einem Antwortfeld unseren Namen einzugeben. (*kurz herzeigen*)

Wir möchten, dass der Hund Hallo und anschließend unseren Namen sagt, wenn wir diesen richtig eingegeben haben. Sollten wir einen neuen Namen eingeben, sagt der Hund "Schön, dich kennenzulernen!".

Hier kommen wir jetzt zu unserem neuen Programmierkonzept, das wir heute lernen: die bedingte Anweisung / Verzweigung.

Wir finden die passenden Blöcke in der Kategorie "Steuerung", aus der wir auch letzte Stunde die Wiederholungsblöcke verwendet haben.

Es gibt zwei Versionen der Bedingung: (beide Blöcke zeigen)

wichtig: **Eine Bedingung ist immer entweder wahr oder falsch.**

Mit dem Block FALLS – DANN kann man eine Anweisung nur dann ausführen lassen, wenn eine bestimmte Bedingung erfüllt ist.

Zum Beispiel: FALLS ich meinen Namen richtig eingegeben habe, DANN sagt der Hund "Willkommen zurück, Verena". Die Bedingung hier ist also, dass ich meinen Namen richtig eingebe.

Anders als bei Wiederholungen werden die Blöcke innerhalb des dann-Abschnitts nur einmal ausgeführt. In jedem Fall läuft das Programm mit der Anweisung weiter, die außerhalb dieses Blocks unten dranhängt.

Mit dem Block FALLS – DANN – SONST kann man zusätzlich festlegen, welche Anweisung oder Sequenz ausgeführt werden soll, wenn die Bedingung nicht erfüllt ist.

Diesen Block wählen wir nun auch für unser Programm aus.

Die Bedingung wird in das eckige Feld nach dem "falls" eingefügt. Wir möchten dabei schauen, ob die eingegebene Antwort mit unserem eigenen Namen übereinstimmt. Dafür können wir aus der Kategorie "Operatoren" den Block mit dem = auswählen.

Wir ziehen dann den Block "Antwort" aus der Kategorie "Fühlen" in das linke Feld und schreiben in das rechte Feld unseren Namen. So kann überprüft werden, ob wir am Ende unseren Namen richtig eingegeben haben. Wenn das der Fall ist, soll der Hund "Willkommen zurück + unseren Namen" sagen.

Frage: Welchen Block müssen wir dafür in den falls-Teil einfügen?



Nun brauchen wir noch den zweiten Teil des Bedingungsblocks. Dieser tritt ein, wenn wir unseren Namen falsch oder etwas anderes eingegeben haben. Wir wollen in diesem Fall, dass der Hund dann zum Beispiel "Schön, dich kennenzulernen!" sagt.

Frage: Welche Blöcke müssen wir dafür einfügen?

Screensharing AUS

Fügt diese Blöcke ebenfalls in euer Programm ein.

Jetzt haben wir die Aufgabe geschafft und ihr könnt ausprobieren einen eigenen Dialog mit der Figur zu führen.

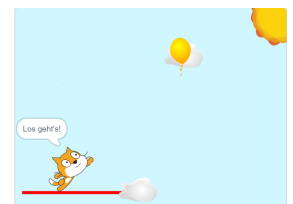


4. Aufgabe 2: Flying Cat

Screensharing AN

Bei diesem Spiel soll die Katze zur Sonne fliegen, ohne dabei von einer der herumschwebenden Wolken berührt zu werden. Die Katze kann über die Pfeiltasten gesteuert werden.

Berührt sie den Luftballon, fliegt sie ein wenig schneller als normal.



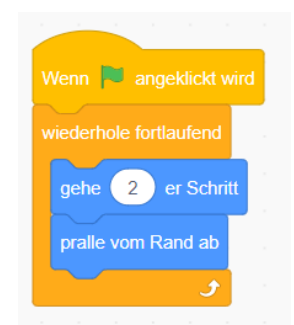
Aufgabe 2.1:

Wir fügen eine Katze, die Sonne, eine Wolke und eine rote Linie hinzu.

Aufgabe 2.2:

Damit die Wolke(n) durch die Gegend schweben, fügen wir als erstes dafür ein paar Anweisungen ein. Dafür brauchen wir einen Wiederholungsblock, wie wir ihn letzte Woche kennengelernt haben. Außerdem müssen Anweisungen eingefügt werden, damit sich die Wolke bewegt und immer wieder vom Rand abprallt.

Frage: Welche Blöcke könnten wir hier einfügen?



Fügt die Figuren ein: die Katze, die Sonne, mindestens eine Wolke und die rote Linie als Startpunkt. Mit der Maus könnt ihr die Figuren dann dort hinziehen, wo es euch passt.

Fügt nun auch selbst dieses Skript mit Wiederholung in der Wolke hinzu.

Screensharing AUS

Aufgabe 2.3:

Screensharing AN

Wir möchten nun mit den Anweisungen für die Katze starten. Als erstes benötigen wir die Pfeilfunktion, damit wir die Katze nach links und rechts mit den Pfeiltasten steuern können.



Frage: Weiß jemand, wie man das machen könnte?

Bitte fügt nun auch selbst diese beiden kurzen Skripte hinzu.

Aufgabe 2.4:

Als nächstes brauchen wir Anweisungsblöcke, die das Programm starten und die Katze an die richtige Stelle (links unten auf die rote Linie) setzen. Wie das geht, wisst ihr schon.

Frage: Welche Blöcke verwenden wir?

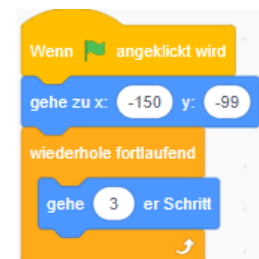
Bitte fügt die entsprechenden Blöcke ein.

Aufgabe 2.5:

Die Katze soll sich durchgehend bewegen.

Frage: Welche Blöcke brauchen wir, damit sich die Katze bewegt und damit dies fortlaufend passiert?

Bitte fügt die richtigen Blöcke nun selbst ein.



Screensharing AUS

Aufgabe 2.6:

Screensharing AN

Sobald die Katze eine Wolke berührt, stürzt sie ab und geht zurück zur Startposition.



Wir haben ja heute die bedingte Anweisung kennengelernt.

Frage 1: Welchen Block könnten wir verwenden, um abzufragen, ob die Wolke berührt wird?

Frage 2: Welche Anweisung müssen wir innerhalb des Bedingungs-Blocks einfügen, damit die Katze wieder zur Startposition zurückgeht?

Fügt nun bitte die entsprechenden Blöcke in euer Programm ein.

Screensharing AUS

Falls noch Zeit ist (mehr als 10 Minuten, Aufgabe 2.7)

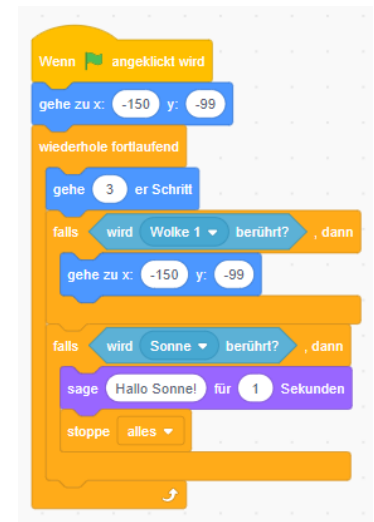
Aufgabe 2.7:

Screensharing AN

Sobald die Katze die Sonne berührt, ist das Spiel gewonnen und das Programm wird beendet.

Frage 1: Was brauchen wir also wieder für einen Block, um diese Bedingung abzufragen? Wo muss der Block eingefügt werden?

Frage 2: Welche Anweisungen müssen innerhalb des Bedingungsblocks verwendet werden, um "Hallo Sonne" zu sagen und danach das ganze Programm automatisch zu stoppen?



Fügt bitte jetzt die entsprechenden Blöcke zu eurem Programm hinzu.

Als Hausaufgabe sollt ihr noch mehr Wolken und deinen Ballon einfügen und Bühnenbildwechsel.

Nächste Woche wird es aufregend mit einem weiteren Spiel.

- - - voraussichtliches Ende der Stunde - - -

Weitere Erweiterung als Hausaufgabe: Wolke 2 und Ballon einfügen

Einfügen von Sprechblase am Anfang; Bühnenbildwechsel (sonst als Hausaufgabe)

4. Stunde: Bedingte Wiederholung

Begrüßung

- *Hallo zusammen. Schön, dass ihr wieder da seid zu unserer vierten Stunde.*
@Beobachter: Teilnehmendenliste checken
- **Kurze** Retrospektive zu letzter Stunde? Fragen nach Hausaufgabe:
Habt ihr eine zweite Wolke und einen Luftballon bei eurem Spiel geschafft?
Habt ihr ein neues Bühnenbild "Gewonnen" kreiert?

Einführung heutiges Thema

- Was erwartet euch heute? Bedingte Wiederholungen. Dafür zeige ich euch kurz ein kleines Spiel, das ihr alle vom Schulhof / aus eurer Freizeit kennt :)

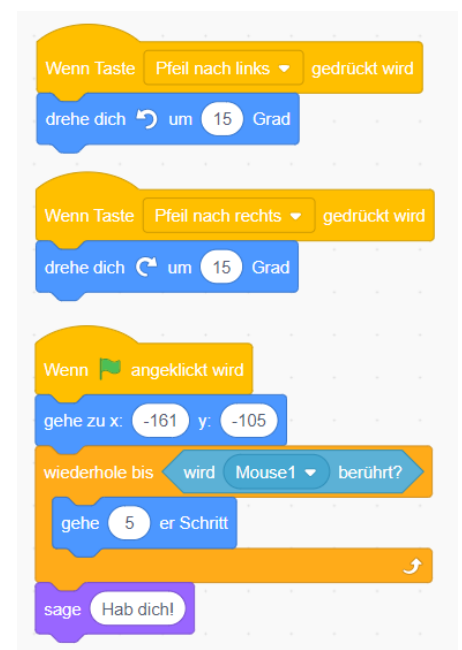
Screensharing AN Spiel zeigen

Aufgabe 1: Fangspiel (Katze soll Maus fangen)

Aufgabe 1.1

Skript **Katze**:

- Wir wollen, dass sich die Katze solange bewegt, bis sie die Maus berührt, also "gefangen" hat. Wir kennen die beiden grundlegenden Konzepte schon aus den letzten beiden Stunden: Wir wollen also eine Sequenz wiederholen (Stichwort Wiederholungen) bis ein Ereignis eintritt, also eine Bedingung uns einen Wert "wahr" liefert. Das heißt, wir brauchen eine **bedingte Wiederholung**, die ihr auch bei den Kontrollstruktur findet.
- Als erstes schreiben wir das Skript für die **Katze**:
- Frage: Was können wir für Blöcke verwenden, um die Katze selbst zu steuern?
→ Wenn Taste 'Pfeil nach links/rechts' gedrückt wird
- Dann brauchen wir noch einen Block, der das Programm startet und die Katze dann zu Anfang nach links unten setzt.
- Frage: Wer kann mir sagen, welche beiden Blöcke ich dafür einfügen muss?
→ grüne Flagge, gehe zu



- Die Katze soll sich jetzt solange bewegen, bis sie die Maus gefangen (also berührt) hat und dann "Hab dich" sagen.
 - Dafür brauchen wir den soeben erklärten wiederhole-bis-Block
 - Frage: Was für einen Block wählen wir als Bedingung im wiederhole-bis-Block? (wird Maus berührt?)
 - Frage: Welche Anweisung müssen wir innerhalb des wiederhole-Blocks einfügen, damit die Katze sich durchgehend fortbewegt? (gehe 5er Schritt)
 - Frage: An welcher Stelle fügen wir den "Sage 'Hab dich'"-Block ein?
 - **Screensharing AUS:** Bitte fügt die besprochenen Blöcke jetzt selbst ein. Erstellt ein neues Projekt und fügt zwei Figuren ein. Ihr müsst nicht die Katze und die Maus nehmen.
 - So, die Anweisungen für die Maus (bzw. für eure "zu fangende"-Figur) könnt ihr jetzt **selbst** erstellen.
- Screensharing AN:** Nochmal herzeigen und kurz erklären, wie Skript von Katze ungefähr aussieht und dass das der Maus so ähnlich sein sollte. (Sicherstellen, dass alle die Katze schon mal fertig/richtig haben).

Screensharing AUS

Aufgabe 1.2

Hinweise in den Chat:

- Am Anfang soll sich die Maus rechts oben befinden
- Die Maus soll sich solange zu zufälligen Positionen bewegen, bis die Katze sie gefangen (also berührt) hat



Alle soweit fertig? Gibt's noch Fragen? (kurz warten) So, dann machen wir das jetzt zusammen, kann mir dabei jemand helfen?

Screensharing AN

Kurze Erklärung:

- Im Skript der Maus fügen wir also fast die gleichen Blöcke wie bei der Katze ein → grüne Flagge, gehe-zu, wiederhole-bis 'wird Katze berührt?', gehe zu Zufallsposition
- Außerdem fügen wir vor dem gehe-zu-Zufallsposition noch einen warte-1-Sek-Block ein, damit die Maus sich nicht zu schnell bewegt und man die Bewegung auch beobachten kann
- nochmal sicherstellen, dass die Kinder das soweit verstanden/erledigt haben (die 4. Stunde fanden alle bisher am schwierigsten)

Screensharing AUS

Aufgabe 1.3

- Ihr dürft jetzt selbst kreativ werden und mehrere Levels mit verschiedenen Aufgaben in euer Spiel einbauen!
- *Das erste Level haben wir ja schon - die Katze fängt die Maus* (bzw. eben mit den Figuren, die ihr selbst ausgewählt habt).
- Damit wir sehen, dass wir ins nächste Level gelangen, möchten wir, dass sich das Bühnenbild ändert, sobald die Maus gefangen wurde. Außerdem soll die Maus dann von der Bildfläche verschwinden und eine neue Figur auftauchen.
- *Im zweiten Level könnten wir dann zum Beispiel einen Dialog einbauen.* Wie das geht, haben wir letzte Woche schon gelernt. Ich zeige euch jetzt nochmal kurz, wie das ganze dann aussehen könnte.

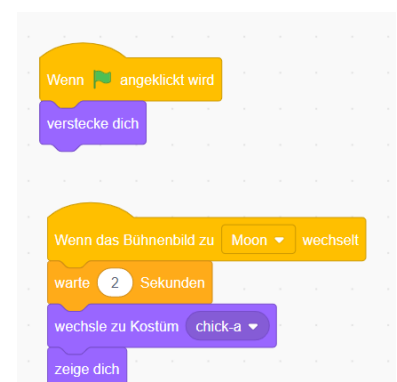
Screensharing AN: Spiel nochmal zeigen mit Bühnenbildwechsel.

Ich gebe euch jetzt noch ein paar Hinweise, damit ihr ungefähr wisst, wie das Ganze funktionieren könnte. Dann habt ihr den Rest der Stunde Zeit selbst aktiv zu sein und uns Fragen zu stellen, falls ihr was nicht verstanden habt oder euer Projekt herzeigen möchtet.

- Maus muss sich verstecken, damit sie in Level 2 nicht mehr zu sehen ist (ihr müsst also am Ende des Skripts von der Maus ein "verstecke dich" einfügen)
- hat-Block: wenn Bühnenbild zu Bühnenbild2 wechselt -> Küken soll sich zeigen

d.h. wenn die grüne Flagge angeklickt wird, soll die Figur aus Level 2 sich noch verstecken und erst, wenn zum Bühnenbild für das Level 2 gewechselt wird, soll sie sich zeigen. Einen geeigneten Block dafür findet ihr in der Kategorie Ereignisse.

- Ihr könnt dann zum Beispiel einen Dialog mit Frage-Block aus der Kategorie Fühlen in Level 2 programmieren. Denkt dabei auch an die Bedingungsblöcke (falls-dann und falls-dann-sonst), die wir vorletzte Woche kennengelernt haben
- Ihr könnt zum Beispiel auch wieder Kostümwechsel und am Ende ein "Gewonnen"-Bühnenbild einbauen. Nun seid ihr aber selbst gefragt, eure eigenen Ideen umzusetzen!



Den Kindern nochmal sagen:

Denkt an:

- verstecke-dich (Kategorie Aussehen)
- Bühnenbildwechsel (Kategorie Aussehen)
- verschiedene Möglichkeiten, ein Skript zu starten (Kategorie Ereignisse)
- der Rest bleibt euch überlassen! Ihr könnt gerne noch mehr Levels (z.B. Fangen, Dialoge, Quiz) hinzufügen oder euer Spiel ausschmücken (z.B. Farbeffekte, Klänge etc.)

Screensharing AUS

Falls noch Zeit ist: Kinder sollen jetzt ihre Projekte weiter bearbeiten (also z.B. Level und Figuren hinzufügen, Farbeffekte etc.) oder auch gerne herzeigen. Das ist komplett ihnen überlassen und ihr könnt eventuelle Fragen beantworten.

Bitte überprüft dabei auch, dass die **Projekte veröffentlicht** sind!

Falls ihr noch nicht fertig seid mit den Levels, ist das Hausaufgabe! Fügt noch mehr Levels ein und Figuren.

Umfrage

- Umfrage
- Vielen Dank fürs Mitmachen. Nächste Woche lernen wir das letzte Programmierkonzept und ihr könnt wieder eigenständig kreativ sein. Es wird *abgespaced* und sehr aktiv!

5. Stunde: Variablen

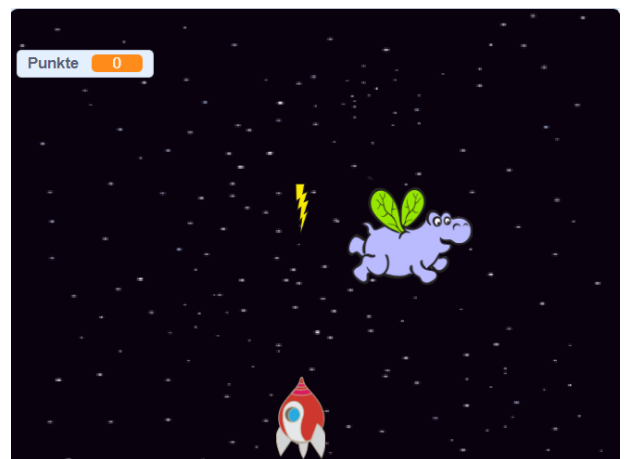
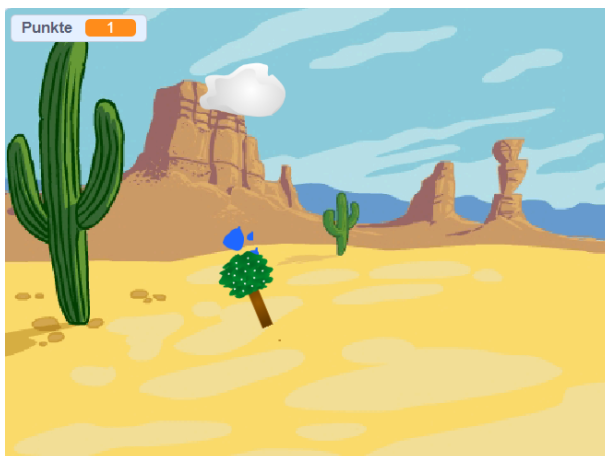
1. Begrüßung

- *Hallo zusammen. Schön, dass ihr wieder da seid zu unserer fünften Stunde.*
@Beobachter: Teilnehmendenliste checken
- Kurze Retrospektive zu letzter Stunde? Fragen nach Hausaufgabe:
Habt ihr noch weitere Level in euer Spiel eingebaut oder euer Spiel noch weiter ausgeschmückt?
- Falls es noch Fragen oder Probleme zu letzter Stunde oder euren Projekten gibt, könnt ihr die gerne später klären!

2. Einführung heutiges Thema

- Damit wir unser Spiel noch spannender machen, wäre es super, wenn wir unsere Punkte speichern und anzeigen lassen könnten und beispielsweise beim Fangen der Maus (wenn ihr an das Spiel von letzter Woche zurückdenkt) den Spielstand erhöhen.
- Das heißt, wir wollen bestimmte Werte speichern. Dafür benutzen wir Variablen.

Screensharing AN: zwei Spiele zeigen: Clone Wars und Green City
(Aktion mit Leertaste: Schießen vs. Gießen; hier seht ihr links oben den Punktestand. Damit wir die Punkte speichern können, benötigen wir eine



Variable. Name der Variable: Punkte,
Wert: z. B. 5, der Wert der Variable kann sich ändern -> er ist VARIABLE)

Bei Green City erwähnen: bei dieser Version haben wir die Tropfen übrigens selbst mit der Funktion "Malen" als Kostüm erstellt. Falls ihr dazu Fragen habt, könnt ihr die später gerne stellen, wenn ihr euer eigenes Programm erstellt.

3. Aufgabe 1: Eigenes kleines Spiel mit Punktestand

- Ähnlich wie zu den beiden gezeigten Spielen, sollt ihr jetzt eure eigenen Figuren und ein Bühnenbild selbstständig hinzufügen.
- Eine Figur (hier die Wolke oder die Rakete) soll dabei mit den Pfeiltasten links und rechts gesteuert werden können. Dafür habt ihr jetzt kurz Zeit. Denkt auch an speichern und veröffentlichen des Projekts!

Screensharing AUS

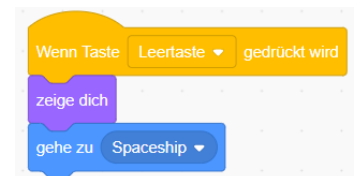
- *Wer möchte denn mal seine Figuren herzeigen?* (von Kindern zeigen lassen, welche Figuren sie gewählt haben)

Screensharing AN:



Ihr solltet nun ein Bühnenbild erstellt haben und außerdem eine Figur und diese Anweisungen hinzugefügt haben, damit die Figur mit den Pfeiltasten bewegt werden kann.

- Wir zeigen vor: wie man schießt/gießt und sich der Punktestand erhöht
- **UMFRAGE:** Welches Spiel möchtet ihr lieber gezeigt bekommen?
- Wir möchten jetzt zunächst noch besprechen, wie wir die Figur schießen bzw. gießen lassen können. Wir möchten, dass geschossen wird, wenn wir die Leertaste drücken. Natürlich brauchen wir dazu erstmal noch eine weitere Figur, die wir als Geschoss verwenden können.
- -> Blitz/Wasser hinzufügen
- *Frage: Welchen Block müssen wir verwenden, damit wir mit der Leertaste schießen können?*



-> wichtig ist dann als nächstes, dass wir den gehe-zu-Block verwenden und damit zu unserer Figur gehen, die schießen soll. Somit sieht es dann so aus, als würde das Geschoss aus der Rakete rauskommen. Außerdem können wir gleich auch einen zeige-dich-Block einfügen, damit der Blitz auch immer zu sehen ist, wenn die Leertaste gedrückt wird.

Screensharing AUS: Fügt den Ereignis- und den gehe-Block in euer Programm ein.

Screensharing EIN



Als nächstes soll sich der Blitz/das Wasser solange nach oben bzw. unten bewegen, bis der Rand berührt wird. Dann soll er verschwinden.

Frage: Welche Blöcke benötigen wir hierfür?

Screensharing AUS: fügt die besprochenen Blöcke nun ebenfalls hinzu.

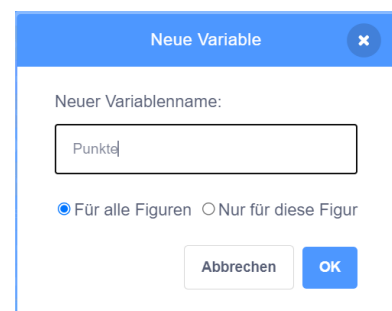


Screensharing EIN: Falls das Nilpferd / der Baum berührt wird, wollen wir, dass der Blitz / das Wasser wieder vom Bildschirm verschwindet.

Frage: Wie könnten wir das machen?

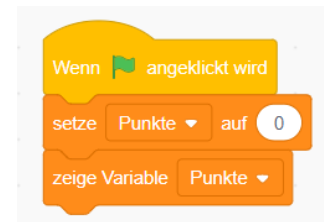
Wichtig: Damit das ganze reibungslos funktioniert, müssen wir in den falls-Block auch ein warte-1-sek einfügen, da die Berührung sonst teilweise nicht lang genug ist und sich der Blitz (bzw. das Wasser) dann nicht versteckt.

- Ziel der heutigen Stunde ist ja außerdem, dass wir auch den Punktestand (und eventuell auch wieder verschiedenen Levels) anzeigen lassen können. Damit das möglich ist, verwenden wir Variablen. Eine Variable dient als Platzhalter für Werte.
- Wir finden diese in der Kategorie "Variablen". Dort können wir eine neue Variable erstellen.
- Ihr könnt euch das wie einen Karton vorstellen, in dem genau ein Gegenstand Platz hat. Wenn wir



den Block Variable für Punkte erzeugen, dann ist das ein Karton auf dem der Variablenname "Punkte" steht. Am Anfang unseres Programms haben wir noch keine Punkte, der Karton ist also noch leer.

Also müssen wir unseren "Punkte" erstmal befüllen, dass nennen wir setzen/initialisieren der Variable und stecken eine Notiz mit dem Anfangswert in den Karton und schließen diesen wieder.



- *Frage: Wir wollen, dass der Punktestand um 1 erhöht wird, sobald der Blitz das Nilpferd abgeschossen/die Wolke den Baum gewässert hat. Welchen Block brauchen wir dafür und an welcher Stelle im Code könnten wir diesen einfügen?*

Screensharing AUS

- Jetzt baut euch euren eigenen Punktestand in eurer Programm ein.

warten

- Wie schaut es bei euch aus? Habt ihr alle einen eigenen Punktestand eingebaut? Ggf. kurz Projekte herzeigen lassen
- Außerdem habt ihr jetzt den Rest der Stunde Zeit, um euer Programm noch zu erweitern (z.B. dass sich das Nilpferd oder der Baum durch die Gegend bewegt).
- *falls Stuttering Movement: gleiche Anweisung mit Maus anstatt Pfeiltasten (Tablet-Kinder sollen anstatt Pfeiltasten Maus/Finger hernehmen)*
- **für Rest der Stunde bzw. Hausaufgabe:** Variablen können auch Text sein, daher könnt ihr z.b. euren Namen vor Beginn des Spiels abfragen. Mehrere Levels eintragen oder verschiedene Leben (Spiel gewonnen, verloren, etc.).
- Falls wir hier schon früher fertig werden, ist das auch okay (dann könnt ihr die Kinder auch 10 Minuten früher gehen lassen)

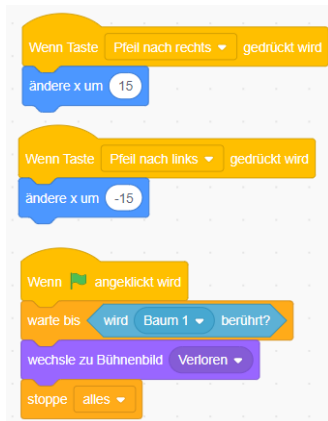
4. Umfrage

- Umfrage
- Vielen Dank fürs Mitmachen. Nächste Woche ist schon unsere letzte Stunde. Falls euch noch was unklar ist oder ihr Fragen habt, dann habt ihr nächstes Mal noch die Gelegenheit dazu.
- Seid gespannt (blabla): Wir haben noch einiges für euch vorbereitet! Schönes Wochenende!



Lösungsskizze

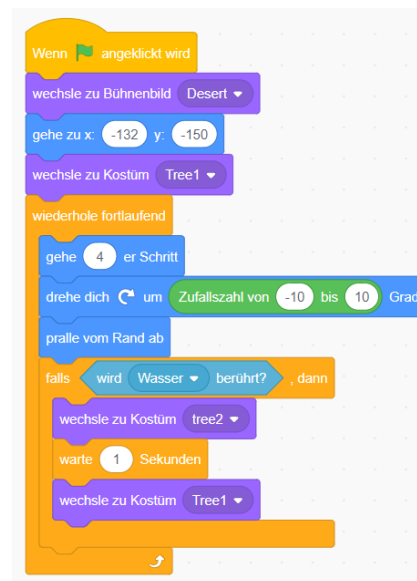
Wolke / Rakete:



Wasser / Blitz:



Baum / Nilpferd:



6. Stunde: Wiederholung der Programmierelemente und eigenes Projekt

1. Begrüßung

- *Hallo zusammen. Schön, dass ihr wieder da seid zu unserer letzten Stunde.*
@Beobachter: Teilnehmendenliste checken
- Kurze Retrospektive zu letzter Stunde? Fragen nach Hausaufgabe:
Habt ihr es noch geschafft, euer Nilpferd/whatever durch den Raum schweben zu lassen?

2. Einführung heutiges Thema

- Heute könnt ihr komplett kreativ euer eigenes Programm erschaffen.
- Dabei sollt ihr die bereits gelernten Programmierelemente anwenden.
- Was habt ihr bereits kennengelernt? (**Programm Goodbye** zeigen, wo die Figuren der letzten Stunden vorbeilaufen)

Screensharing AN

- 1. Stunde: Grundstrukturen (Zauberer und Hase)
-> Erweiterung Text zu Sprache kennengelernt
- 2. Stunde: Wiederholungen (Kontrollstruktur) (Eisbär und Pinguin, Tanz) -> Erweiterung Malen
- 3. Stunde: Bedingte Anweisungen: falls -- dann, --; verschiedene Startbausteine (hat-blöcke, Ereignisse), operatoren (Flying Cat)
-> Dialog zwischen Hund und Katze; Spiel Katze zu Sonne fliegen, ohne die Wolken zu berühren, Steuerung mit Tasten
- 4. Stunde: Kontrollstrukturen (Steuerung): bedingte Wiederholung -> Katz-Maus-Fangspiel
- 5. Stunde: Variablen -> Werte erzeugen und verändern, Hippo Spiel

3. Heutige Aufgabe

Wir zeigen euch kurz ein **Spiel** (Räume wechseln, "Räume"/"Meine Welt").

Ihr sollt heute auch versuchen, ein Spiel zu bauen, in dem es verschiedene Räume bzw. Level gibt, in die man durch bestimmte Objekte, wie bspw. Türen, an den Seiten hineingehen kann. Jeder Raum ist dann ein anderes Level.

Das einzige, was ihr beachten sollt, ist, dass ihr die Elemente aus den letzten Stunden verwenden sollt (auch in den Chat schreiben):

- Klänge und Aussehen

- Wiederholung
- Bedingung (falls - dann)
- bedingte Wiederholung
- Variablen

Eurer Kreativität sind aber sonst keine Grenzen gesetzt, ihr könnt das Ganze z.B. auch als Unterwasserwelt, in der Stadt, im Weltraum oder wie ihr wollt gestalten! Unser Spiel ist nur ein Beispiel, falls ihr noch etwas Orientierung oder Hilfe braucht.

- Wir werden das Projekt jetzt gemeinsam beginnen und dann habt ihr den Rest der Stunde Zeit, um daran selbst weiterzuarbeiten.
- Falls ihr Fragen habt zu einem Programmierkonzept oder Hilfe benötigt, dann hebt einfach während der Stunde eure Hand.
- Ihr könnt euch inhaltlich total austoben, wichtig ist nur, dass ihr die eben genannten Elemente in eurem Programm anwendet.

Aufgabe 1.1

Erstellt ein neues Projekt, gebt diesem einen Namen (bei uns z.B. "Meine Welt") und veröffentlicht es.

Jetzt brauchen wir verschiedene Räume. Wir erstellen erstmal zwei verschiedene, ihr könnt dann später noch so viele weitere hinzufügen wie ihr möchtet.

Ich möchte euch hierbei nochmal zeigen, dass man auch auf den Bühnenbildern malen kann. Wählt dazu einfach das aktuelle Bühnenbild aus und geht dann auf Bühnenbilder. Hier könnt ihr jetzt zum Beispiel auch selbst bunte Balken erstellen oder mit dem Stift zeichnen. (herzeigen)

Erstellt jetzt bitte zwei verschiedene Bühnenbilder. Wählt einen Hintergrund der euch gefällt und fügt dann bei dem ersten Bühnenbild auf der rechten Seite einen Balken oder eine Figur hinzu, die ihr als Tür verwenden wollt. Im zweiten Bild fügt ihr eine andere Farbe/Figur auf der linken Seite hinzu.

Außerdem könnt ihr jetzt auch schon eine Figur hinzufügen, die sich dann durch die verschiedenen Räume bewegen soll. In unserem Beispiel ist das der Ball.

Fügt die entsprechenden Befehle hinzu, damit sich die Figur mit den Pfeiltasten nach oben, unten, links und rechts bewegen lässt.



Denkt daran, am besten die “ändere x bzw y um ...”-Blöcke zu verwenden. Das kennt ihr bereits von den letzten beiden Stunden (ggf. Erinnerung an das - wenn nach links gegangen werden soll).

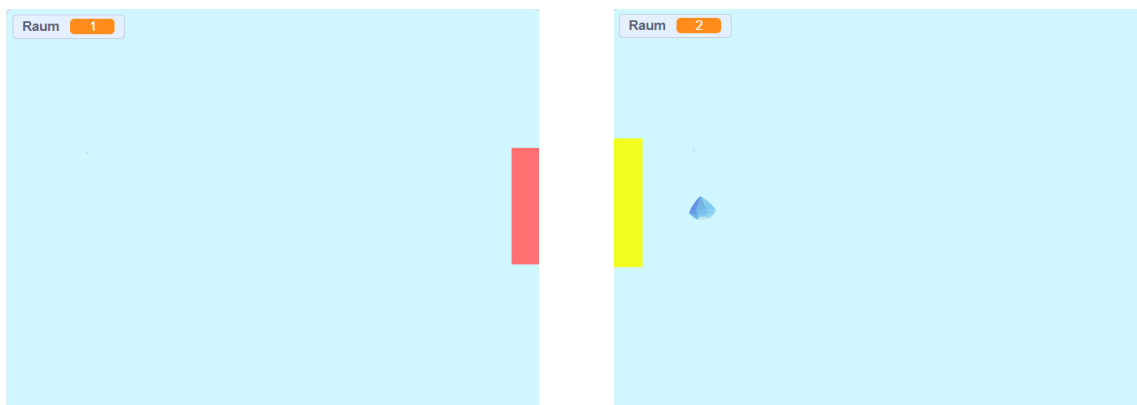
SCREENSHARING AUS

Hat das soweit jeder geschafft?

SCREENSHARING EIN

Aufgabe 1.2

Als nächstes wollen wir noch eine Variable hinzufügen, die uns immer anzeigt, in welchem Raum wir uns befinden. Die Variablen kennt ihr ja aus letzter Stunde.



Erstellt eine neue Variable mit dem Namen “Raum” oder “Level”.

Fügt jetzt bei eurer Figur die grüne Flagge ein und eine Anweisung, die am Anfang die Variable Raum auf 1 setzt. Außerdem den Block, der das erste Bühnenbild festlegt. Das kennt ihr auch schon aus letzter Stunde.



SCREENSHARING AUS

Hat das soweit jeder geschafft?

SCREENSHARING EIN

Aufgabe 1.3

Nun wollen wir zwischen den Räumen hin und herwechseln können.

(Tipp: Also 'falls' diese Farbe berührt wird, soll sich unser Bühnenbild wechseln)

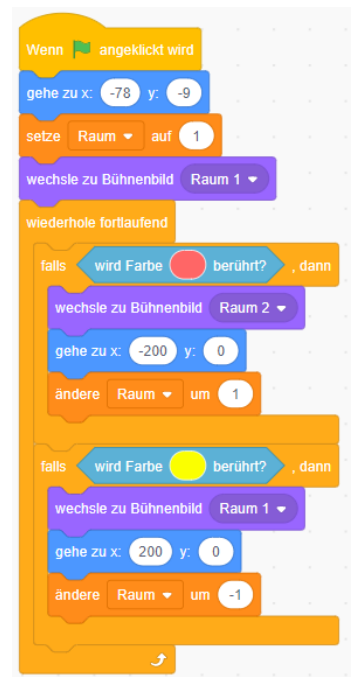
Frage: Wie könnten wir das machen? Welche Blöcke?

-> Wir verwenden einen **falls-Block** und überprüfen, ob die Farbe oder die Figur berührt wird. Wenn dies der Fall ist, wechseln wir zum nächsten Bühnenbild. Das Ganze muss sich in einer fortlaufenden Wiederholung befinden, damit die Bedingungen durchgehend überprüft werden.

Frage: Wie muss unsere Variable verändert werden und an welchen Stellen?

-> ändere Raum-Variable um +/- 1

-> außerdem können wir ein gehe-zu einfügen, damit die Figur an der richtigen Stelle wieder rauskommt.



So, damit kommt ihr schon mal in den zweiten Raum.

Falls ihr in den nächsten Raum möchtet, funktioniert das ganz genauso wie wir das gerade gemacht haben.

Falls ihr wieder zurück in den ersten Raum oder das erste Level möchtet, funktioniert das ähnlich, nur, dass sich euer Raum nicht um eins erhöht, sondern eins abgezogen wird (also -1).

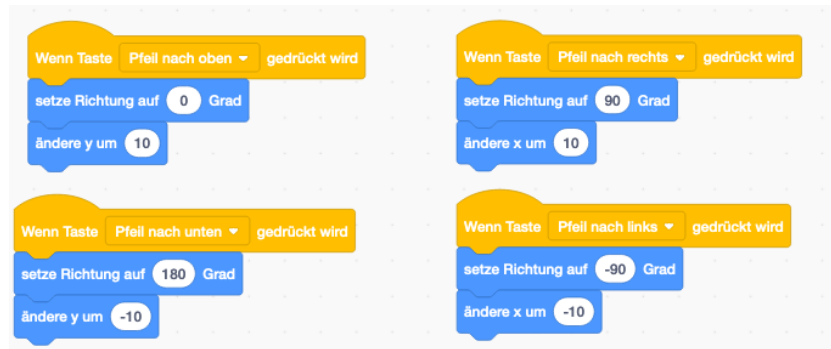
Das könnt ihr jetzt umsetzen.

Also schaut, dass ihr eure Räume oder Level in eurer Welt wechselt und fügt z.B. Figuren ein, mit denen man reden kann wenn man sie berührt wie beispielsweise unsere Tafel oder wie unser Käfer, bei dem man verloren hat, wenn man ihn berührt. Erschafft euch einfach eure eigene kleine Welt wie sie euch gefällt und wenn ihr Fragen habt, sind die Betreuer für euch da!

Genug Zeit einplanen, damit Kinder Projekte herzeigen können. Spätestens 10 Minuten vor Ende solltet ihr aber mit der Umfrage beginnen, falls die Kinder noch Fragen danach haben.

Anmerkung:

- für die Hindernisse (Labyrinth): Richtung setzen, damit Figur abprallt



4. Abschluss

- Wir haben uns gefreut.
- Letzte Umfrage
- Ihr könnt mit euren Accounts weiterhin Scratch nutzen und weitere Projekte veröffentlichen. Falls euer Account die nächsten 6 Wochen inaktiv ist, setzen wir diesen wieder zurück.
- Wir haben jeden ersten Freitag im Monat das "Coder Dojo", für diejenigen von euch, die noch weiter Scratch programmieren wollen. Da werden keine neuen Konzepte gelernt, sondern Programme vorgestellt und Hilfestellungen gegeben. Weitere Infos und Kurse gibt es immer wieder auf unserer Webseite. Im Herbst wird es einen fortgeschrittenen Kurs geben.
- Wir möchten uns nochmals bedanken für eure Teilnahme! Wir finden, ihr habt das alle super gemacht, obwohl es virtuell war!
- Schönes Wochenende! Falls noch jemand dringende Fragen hat, wir sind die nächsten 10 Minuten noch für euch da :)