

TUTORIAL ZU SCRATCH 2.0

In diesem Tutorial sollst Du die grundlegenden Funktionen von Scratch ausprobieren und kennenlernen. Danach solltest Du gut mit dem Programm umgehen können und eigene Programme in Scratch schreiben können.

Probiere auch während dem Tutorial immer wieder Funktionen aus, die Du interessant findest. Bitte stelle auch während dem Durcharbeiten dieses Tutorials immer Fragen, falls Du etwas nicht verstehen solltest.

Inhalt

Bei Scratch anmelden	1
Entwicklungsumgebung	3
Das 1. Programm: Bewegungen	4
Klang	6
Aussehen	6
Senden und Empfangen	7
Fühlen und zählen	9
Daten (Variablen und Listen).....	10
Spielerweiterungsidee für besonders Schnelle oder zum Ausprobieren für Zuhause.....	11

Bei Scratch anmelden

Damit du Scratch verwenden und deine Projekte auch gleich veröffentlichen und mit anderen teilen kannst brauchst du ein Scratch-Konto.

Starte deinen Browser. Gehe auf <http://scratch.mit.edu/> und wähle dort .

Damit du dir nicht erneut einen Benutzernamen und ein Kennwort merken musst, verwende für die Anmeldung beim Benutzernamen jeweils die ersten beiden Buchstaben deines Vor- und deines Nachnamens und als Kennwort das gleiche, das du auch für den Educanet2-Zugang verwendest.

Scratcher werden

Es ist einfach (und kostenlos!), sich bei Scratch zu registrieren.

Wähle einen Scratch-Benutzernamen

Wähle ein Kennwort

Kennwort bestätigen



1 2 3

Benutze nicht deinen echten Namen

Entsprechend nutze deine Educanet2-Emailadresse für die Registrierung:

Scratcher werden

Deine Antworten auf diese Fragen werden vertraulich behandelt.


Warum wir nach diesen Informationen fragen ?

Geburtsmonat und -jahr

Geschlecht Männlich Weiblich

Land

E-mail-Adresse

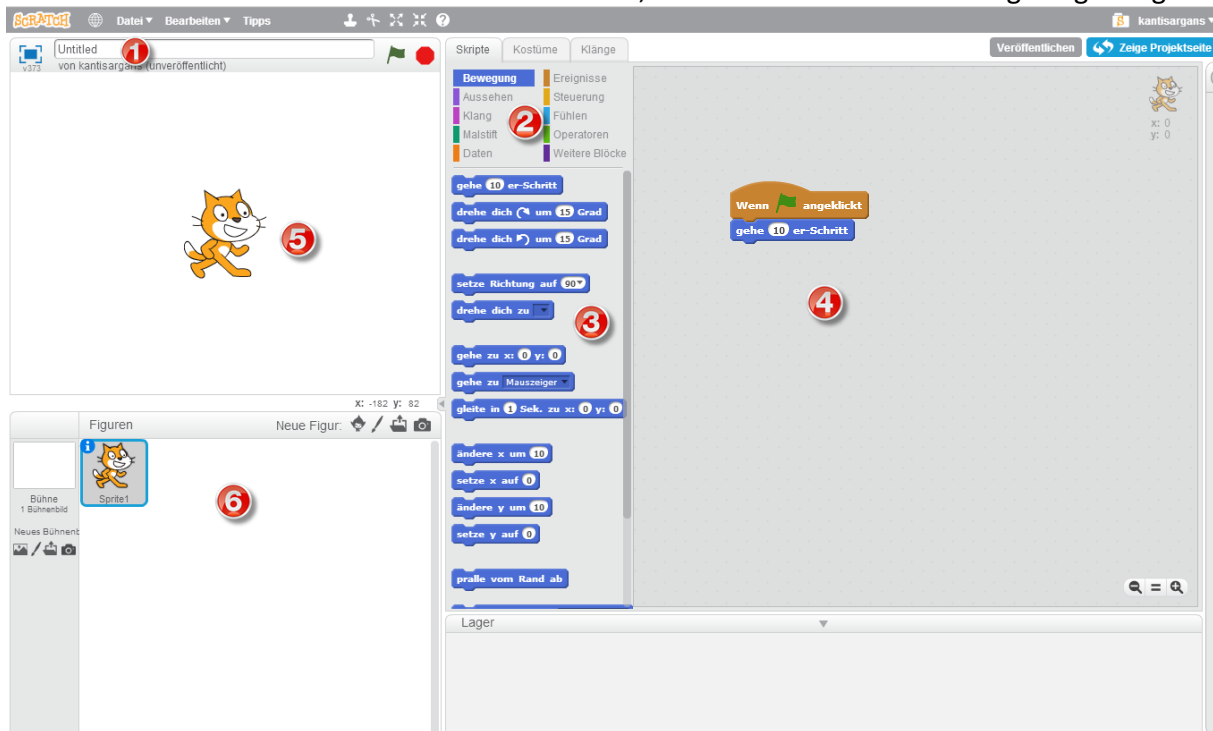


1 2 3

Und schon kann es losgehen!

Entwicklungsumgebung

Sobald du auf die Schaltfläche **Entwickeln** klickst, landest du in der Entwicklungsumgebung:



Legende:

1. Du kannst deinem Projekt einen Namen geben. Es wird automatisch unter diesem Namen gespeichert. Über «Datei»→«Jetzt speichern» kann das Projekt manuell gespeichert werden.
2. Alle Befehle, welche beim Programmieren verwendet werden können, sind in «Blöcke» geordnet.
3. In jedem Block hat es verschiedene Befehle. An der Farbe erkennt man die Zugehörigkeit zum entsprechenden Block.
4. Durch Ziehen der Befehle in den Skript-Bereich wird das Programm erstellt.
5. Auf der Bühne werden die Befehle des Programms ausgeführt.
6. Im Bereich der Figuren findet man alle Objekte, welche im Programm verwendet werden.

Das 1. Programm: Bewegungen



In diesem Kapitel, soll der Umgang mit Bewegungsfunktionen in Scratch erlernt werden. Wir bedienen uns dazu der direkt nach dem Programmstart von Scratch dargestellten Katze und wollen versuchen, ihr Bewegung beizubringen.

Aufgabe: Versuche zunächst, die Katze von links nach rechts bewegen zu lassen.

Ziehe aus dem „Bewegung“-Block die Funktion **gehe 10 er-Schritt** in das Skriptfenster. Die Katze sollte sich nun in 10er Schritten von links nach rechts bewegen, sobald du einen Mausklick auf den Befehl machst.



Um die Bewegung allerdings zu starten benötigst du eine Startbedingung. Diese findest Du im „Ereignisse“-Block.

Wähle hier eine geeignete Startbedingung und verbinde sie durch Ziehen ins Skriptfeld mit der Vorwärtsbewegung.

Wenn du alles richtig gemacht hast, so sollte dein Programm dann so aussehen:



Wenn Du jetzt auf die Grüne Flagge klickst, wirst du sehen, wie sich die Katze langsam vorwärts bewegt. Diesen Ablauf wollen wir allerdings noch deutlich verbessern.

Aufgabe: Probiere auch die rechts abgebildeten «Startbedingungen» mit der Vorwärtsbewegung zu verknüpfen, und teste, wie sie funktionieren.



Warum bewegt sich die Katze eigentlich von links nach rechts? Am unteren rechten Rand der Bühne siehst du etwa Folgendes: x: 40 y: 17

Der „x“ und „y“ Wert gibt die Position der Katze in der Bühne an. Wenn Du mit der Maus über die Bühne fährst, siehst du direkt unterhalb der Bühne die x, y Koordinaten der Mausspitze auf der Bühne.

Aufgabe: Finde heraus, welche Koordinaten die Mitte der Bühne hat. Kannst du auch sagen, welche Koordinaten, das rechte obere Eck hat?

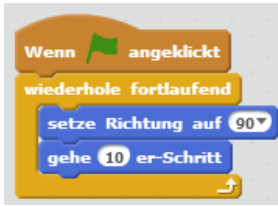
Versuche nun ein Programm zu schreiben, in dem die Katze sich 10 Schritte nach oben bewegt, wenn du auf die grüne Flagge drückst! Dabei kann der Befehl **setze Richtung auf 90** sehr hilfreich sein.



Durch Setzen von Haken bei **x-Position**, **y-Position** und **Richtung** kannst Du Dir laufend Position und Richtung der Katze auf der Bühne anzeigen lassen.

Bis jetzt musst Du, um die Maus sich einmal bewegen zu lassen immer wieder auf die grüne Flagge drücken. Versuche jetzt die Katze ganz nach rechts bewegen zu lassen. Verwende hierfür das Konzept der Wiederholungsanweisung (Schleife). Alles was in der „wiederhole fortlaufend“ – Klammer eingeschachtelt ist, wird immer wieder ohne Ende ausgeführt.

Eine mögliche Lösung könnte so aussehen:



(Durch Drücken der roten Flagge wird die „wiederhole fortlaufend“-Schleife abgebrochen.)

Aufgabe: **Probiere aus, was passiert, wenn Du nachdem die Katze ganz rechts angekommen ist, wieder auf die grüne Flagge drückst.**

Du kannst die Katze mit der Maus wieder an eine andere Position in der Bühne ziehen, um sie erneut nach Drücken der grünen Flagge in Bewegung zu sehen.

Noch schöner ist es allerdings die Katze automatisch an eine Startposition gehen zu lassen, bevor sie mit der Bewegung beginnt. Ergänze dafür den Anfang des Programms um eine Startposition, die die Katze einnimmt, bevor sie beginnt, sich zu bewegen. Verwende hierfür z. B. die Anweisung



Aufgabe: **Finde heraus, wie sich die Geschwindigkeit der Bewegung der Katze beeinflussen lässt! Programmiere die Katze nun so, dass sie langsam dem Mauszeiger folgt.**

Nun soll die Katze, wenn sie auf den Rand der Bühne trifft, in eine andere Richtung weiterlaufen, und so lange durch die Bühne laufen, bis die rote Flagge gedrückt wird.

Versuche z.B. die Funktion zu verwenden.

Aufgabe: **Mache Dich selbst, bevor Du mit der nächsten Lektion beginnst, mit den restlichen Bewegungsoptionen vertraut.**



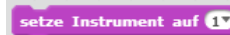
Klang

Nun wollen wir uns ein wenig der Musik zuwenden und uns dazu den „Klang“-Block ein wenig genauer ansehen. Du kannst mittels dieser Befehle Klänge erzeugen.

Schreibe zunächst ein einfaches Programm, dass beim Drücken der grünen Flagge „miau“ macht.



Du kannst aber auch viel kompliziertere Lieder spielen lassen. Wähle durch die Option



ein anderes Musikinstrument und lasse es dann eine bestimmte Note



spielen.

Aufgabe: Lasse doch mal ein Klavier folgende Noten spielen
CDEFGAAAAGAAAAGFFFFEEDDDDC
 Wie heißt das Lied? Hinterlege das doch mal mit einem Schlagzeug!



Aussehen

Die Bewegung unserer Katze sieht leider immer noch nicht ganz gut aus. Das wird jetzt geändert!

Mit der Registerkarte kannst Du der Katze ein anderes Aussehen verleihen. Es gibt von unserer Katze noch ein zweites Bild, das sie zeigt, wenn sie gerade einen Schritt ausführt. Wenn wir die beiden Bilder während der Vorwärtsbewegung abwechseln, können wir den Anschein erzeugen, als ob die Katze geht.

Verwende hierzu das bereits geschriebene Vorwärtsprogramm:



und ergänze es um einen Kostümwechsel. Drücke auf das Register «Kostüme». Hier siehst du alle momentan verfügbaren „Kostüme“ der Katze. Du kannst sie natürlich jederzeit ergänzen, wenn Dir welche fehlen.

Damit die Katze nun tatsächlich ihr Kostüm wechselt musst du dein Skript um einen Befehl aus dem „Aussehen“-Block erweitern.

Hier findest du Befehle, mit denen deine Katze etwas sagen kann, aber auch einen Befehl, mit dem die Katze ihr nächstes Kostüm anziehen kann.

Ergänze dein Programm wie folgt:



Nach dem Drücken der grünen Flagge siehst nun leider, dass die Kostümwechsel im Vergleich zur Bewegungsgeschwindigkeit viel zu schnell sind. Füge daher noch eine kleine Pause nach dem Kostümwechsel ein und betrachte das Ergebnis.



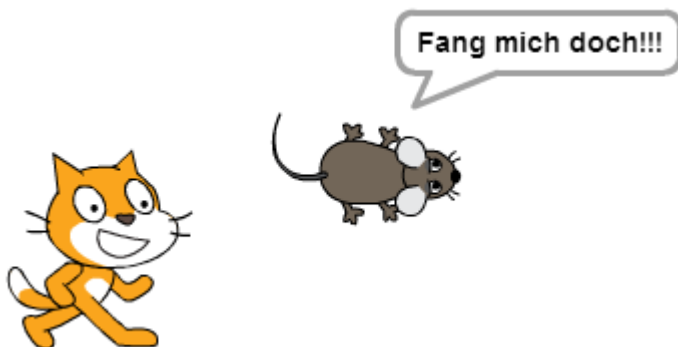
(Tipp: Du findest die verwendeten Symbole immer unter dem Menü, dass die entsprechende Farbe trägt. Die verwendete (gelbe) Pause findest Du also im Steuerungsblock).

Senden und Empfangen

Ein wichtiges Konzept in Scratch ist das „Senden und Empfangen“ von Nachrichten. Das schauen wir uns jetzt noch in einem größeren Programm an.


Mit „Senden und Empfangen“ kann eine Aktion bei einem anderen Objekt ausgelöst werden. Es muss beispielsweise nicht die grüne Flagge gedrückt werden um eine Bewegung bei einer Figur auszulösen, sondern eine Figur muss einer anderen Figur eine geeignete Botschaft senden, damit sie sich bewegt.

Im folgenden schreiben wir ein Programm, in dem eine freche Maus von einer Katze davonläuft, und der Katze immer wieder „Fang mich doch!!!“ hinterher ruft.



Drücke im Menü auf *Datei / neu* und beginne ein neues Programm.

Bevor du nun die Katze programmieren kannst, benötigen wir zunächst eine Maus:

Wähle dazu den **ersten** Knopf in der Leiste **Neue Figur**  und wähle das Bild mit der Maus aus. Mit der Schaltfläche «Verkleinern» in der oberen Menüleiste



kannst du die Größe der Maus nach deinen Wünschen verändern.

Kommen wir nun wieder zur Programmierung der Katze. Klicke dazu auf die Katze, so dass sie wie rechts dargestellt mit einem Rahmen angezeigt wird.



Verwende für die Programmierung des Skripts (\cong Folge von Befehlen) der Katze aus dem „Ereignisse“-Block den Befehl

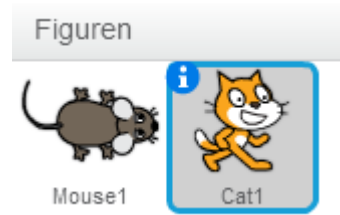
. Erstelle im DropDown-Menü eine neue Botschaft, auf die die Katze reagiert.

Die eigentliche Bewegung kannst Du so übernehmen, wie sie vorher schon mehrfach verwendet wurde. Das Skript der Katze könnte also so aussehen:

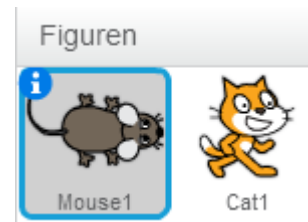
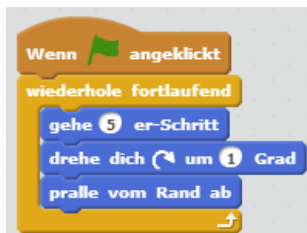


Falls Du auf die grüne Flagge drückst, passiert nun erst einmal gar nichts. Die Katze reagiert nur, wenn ihr die Maus eine Botschaft schickt!


Kommen wir deshalb nun zur Programmierung der Maus. Klicke auf das Bild der Maus, so dass die Skripte der Maus angezeigt werden. Unten rechts sollte nun die Maus mit einem Rahmen dargestellt werden.



Schreibe nun zunächst ein Skript, dass die Maus auf der Bühne herumlaufen lässt. Es könnte so aussehen:



(Die Drehung ist eingefügt, damit die Maus nicht immer nur von links nach rechts läuft).

Wenn du das Programm startest, siehst Du wie die Maus läuft, die Katze aber noch nicht reagiert. Dafür schreiben wir der Maus ein weiteres Skript, in dem die Maus die Katze anstachelt. Verwende hierfür den Befehl . Immer wenn die Maus diese Botschaft sendet, wird die Katze darauf reagieren.

Damit die Botschaft allerdings nicht nur einmal gesendet wird (und die Katze nur einmal hinter der Maus herjagt), solltest Du sie in eine „wiederhole fortlaufend“-Schleife einpacken. Das zweite Skript der Maus könnte also so aussehen:



In einem dritten Skript könntest Du nun die Maus auch noch rufen lassen „Hey Katze! Fang mich!“.


Wenn Du nun die grüne Flagge drückst, werden alle 3 Skripte der Maus gleichzeitig ausgeführt. Zusätzlich reagiert die Katze auf die lebensmüde Maus und jagt diese.



Fühlen und zählen



Bisher merkt die Maus noch nichts davon, wenn sie von der Katze gefangen wird. Das wollen wir jetzt ändern! Dazu brauchen wir zwei Anweisungen aus dem „Steuerung“-Block: nämlich die



Diese führen dazu, dass immer dann, wenn die Bedingung, die in dem Kästchen  angegeben ist, erfüllt wird, die nachfolgenden Anweisungen ausgeführt werden. Ein etwas unsinniges Beispiel könnte so aussehen:



In diesem Beispiel wird also im Falle, dass $3 = 3$ ist, und das ist ja immer der Fall, „Ich liebe Informatik!“ gesagt.

Wie du vielleicht schon bemerkt hast kann die Bedingung nach dem „falls“, die in das Kästchen  gesetzt werden muss, entweder aus dem Block **Operatoren** (siehe Beispiel) oder aus dem Block **Fühlen** kommen. Damit die Katze merkt, dass sie die Maus gefangen hat, können wir die Bedingung  aus dem „Fühlen“-Block verwenden.

Aufgabe **Programmiere die Katze so, dass immer wenn sie die Maus fängt sie für eine bestimmte Zeit „Hab dich!“ sagt.**

Im Folgenden wollen wir jetzt die Maus nicht mehr selbständig über den Bildschirm laufen lassen, sondern wir wollen sie mit der Tastatur steuern.

Dafür können wir aus dem „Fühlen“-Menü die Bedingung  verwenden.

Aufgabe: **Versuche also nun die Maus so zu programmieren, dass immer dann, wenn die Taste „oben“ / „unten“ / „links“ / „rechts“ gedrückt wird, die Maus ihre Bewegung genau in diese Richtung ändert. Es sollen also jetzt nur noch Bewegungen vertikal und horizontal zum Bildschirm möglich sein.**

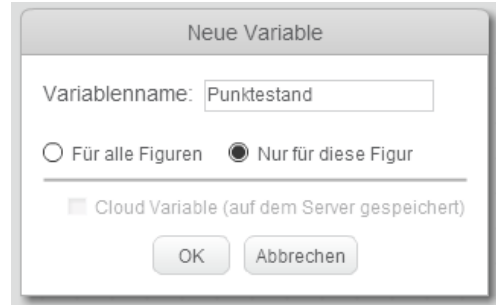
Um unser kleines Katz und Mausspiel noch etwas interessanter zu gestalten wollen wir der Maus jetzt noch ein Ziel geben: Die Maus hat Hunger und will Käse essen.

Aufgabe: **Erstelle ein neues „Käse“-Objekt auf dem Bildschirm. Wie ein solches Objekt entworfen wird, weißt du bereits. Eventuell musst du dir aus dem Internet ein Bild mit einem Käse suchen.**

Daten (Variablen und Listen)

Um ein richtiges Spiel daraus zu machen, wollen wir jetzt der Maus, immer wenn sie den Käse erreicht, Punkte geben. Dazu muss der aktuelle Punktestand in irgendeiner Form gespeichert werden. Um solche zusätzlichen Informationen zu speichern, eignet sich der „Daten“-Block.

Aufgabe: Erzeuge also eine neue Variable mit dem Namen Punktestand, der zunächst nur für die Maus gelten soll. Klicke dazu die Option „Nur für diese Figur“ an. Beachte, dass es möglich ist, die Variablenwerte auf dem Server zu speichern. Dies macht es möglich, z.B. einen Highscore zu machen oder eine Umfrage.



Schau dir dazu das Video auf


<http://www.youtube.com/watch?v=qDFY4O2JU9U#t=106> ab 1'45" an.

Aufgabe: Programmiere die Maus jetzt so, dass immer, wenn der Käse berührt wird, der Punktestand um eins hochgezählt wird und die Maus wieder auf eine Anfangsposition zurückgesetzt wird (wie das mit dem Hochzählen funktioniert solltest du herausfinden können, wenn du den Block Daten genauer ansiehst). Achte darauf, dass der Punktestand, beim Neustart des Programms wieder auf Null gesetzt wird.

Damit die Katze auch ihr Erfolgserlebnis bekommt, soll jetzt immer, wenn sie die Maus gefangen hat, der Punktestand um Eins erniedrigt werden, und die Maus erneut auf ihre Anfangsposition gesetzt werden. Eventuell musst du dafür die Variable neu erstellen, und auf die Einstellung „Für alle Figuren“ setzen, damit auch die Katze direkten Zugriff darauf hat. Vielleicht findest du aber auch eine andere Lösung.

Aufgabe: Ergänze dein Spiel entsprechend.

Damit jetzt das Spiel nicht immer gleich abläuft, die Katze, die Maus und der Käse also nicht immer an derselben Position starten, sollst du jetzt noch das Konzept der Zufallszahlen kennen lernen.

Unter dem „Operatoren“-Block findest du einen Eintrag 

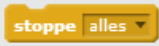
Damit kannst du den Computer eine Zufallszahl in dem von dir angegebenen Bereich erzeugen lassen.

Die Einstellung „Zufallszahl von 1 bis 10“ erzeugt also eine Zahl, die zwischen 1 und 10 liegt. Den Bereich, in dem diese Zahl liegen soll kannst du natürlich durch Angabe von neuen Zahlen in den Kästchen ändern.

Aufgabe: Programmiere das Spiel nun so, dass Katze, Maus und Käse am Anfang des Spiels auf eine zufällige Position gesetzt werden. Ebenso sollen alle Figuren auf eine zufällige Position gesetzt werden, wenn die Maus den Käse erreicht, oder die Katze die Maus fängt.

Kleiner Tipp: du wirst dafür wahrscheinlich wieder Nachrichten senden und empfangen lassen müssen

Was uns jetzt noch fehlt ist ein Ende des Spiels. Um das gesamte Programm zu stoppen gibt es unter

dem „Steuerung“-Block einen Eintrag .

Aufgabe: Erweitere das Programm nun so, dass wenn die Maus 10 Punkte bzw. –10 Punkte erreicht hat, das Spiel beendet wird. Du kannst natürlich die Maus der Katze bzw. die Katze der Maus am Ende des Spiels noch etwas sagen lassen, je nachdem wer gewonnen hat.

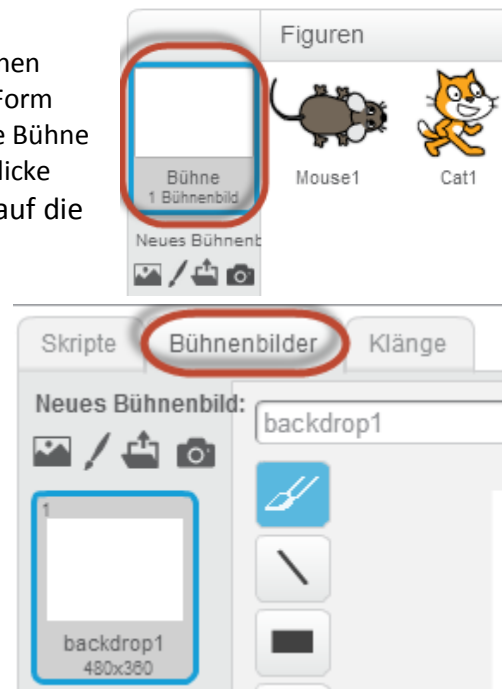
An sich sind wir jetzt am Ende unserer Einführung in Scratch und du solltest dich jetzt mit den wichtigsten Funktionen auskennen. Wenn du noch Zeit hast kannst du jetzt noch selbst Ideen in das Spiel einbringen, oder du bearbeitest noch den Punkt „Spielerweiterungsidee für besonders Schnelle“.

Spielerweiterungsidee für besonders Schnelle oder zum Ausprobieren für Zuhause

Um das ganze Spiel noch auszubauen, wollen wir jetzt einen richtigen Level entwerfen und zwar soll dieser Level die Form eines rechteckigen Labyrinths haben. Zeichne also für die Bühne einen eigenen Hintergrund, der ein Labyrinth darstellt. Klicke dafür zunächst auf das Symbol für die Bühne und dann auf die Registerkarte für die Bühnenbilder

Im Folgenden soll nun der Käse an einem bestimmten Ort, die Maus an einem bestimmten Ort und die Katze an einem bestimmten Ort im Labyrinth platziert werden.

Ziel ist nun erneut, dass die Maus versucht, an den Käse zu kommen, ohne vorher von der Katze erwischt zu werden. Wichtig ist jetzt, dass die Katze und die Maus, wenn sie gegen eine Wand des Labyrinths stoßen, natürlich nicht weiterlaufen können sollen, sondern genau in die Richtung aus der sie kamen zurücklaufen sollen.



Dafür kann für dich die Bedingung **wird Farbe berührt?** von Nutzen sein (Immer wenn also die Farbe der Labyrinthwand berührt wird, soll die Katze/Maus in die Richtung zurücklaufen, aus der sie ursprünglich kam.)

Ausserdem macht es jetzt keinen Sinn mehr, die Katze immer nur in Richtung der Maus laufen zu lassen, da sie sonst ständig gegen irgendwelche Mauern stößt. Vielleicht hast du eine gute Idee, wie du eine „intelligente“ Katze programmieren kannst, die trotz der vielen Hindernisse es noch schafft die vom Menschen gesteuerte Maus effektiv zu verfolgen. Beim Erreichen des Käses könnte dann zum Beispiel ein neuer Level mit einem neuen Labyrinth beginnen. Aber wie du diese Spielidee jetzt genau umsetzt sei ab dieser Stelle dir überlassen...

Viel Spaß beim Programmieren!