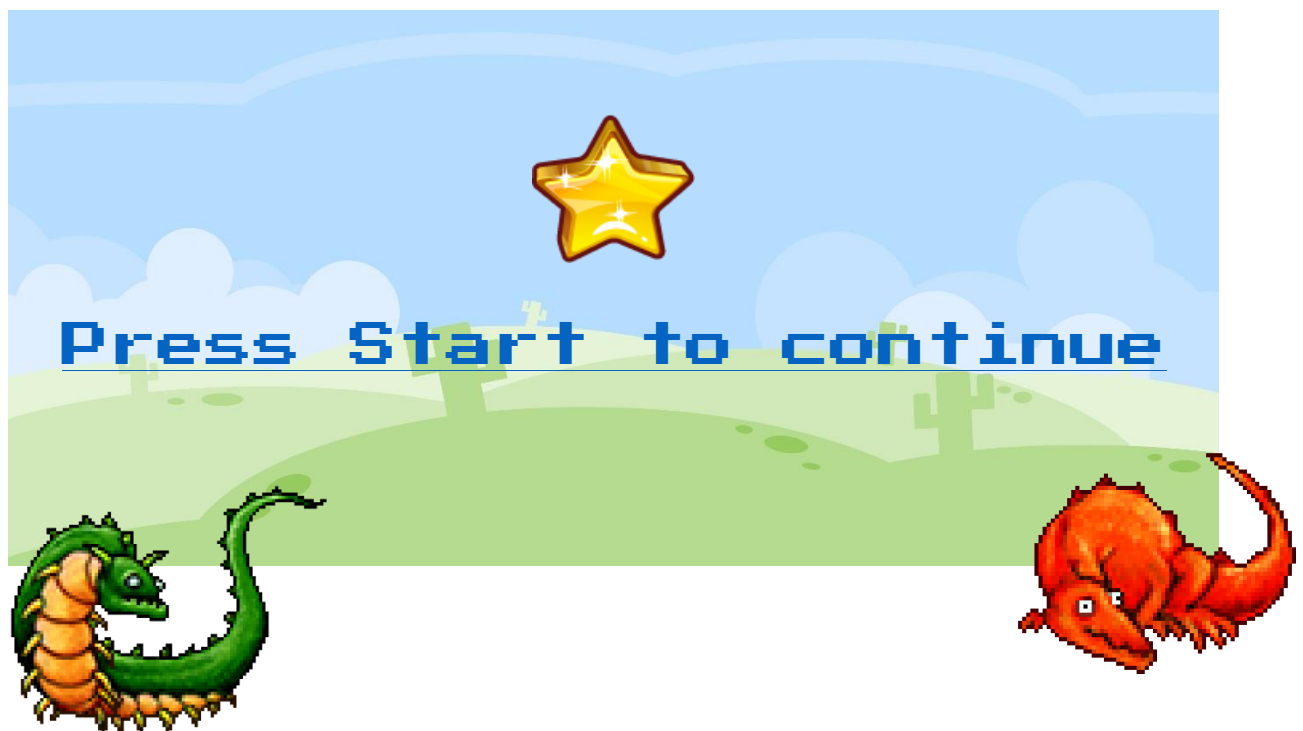




Game Design mit Scratch



Game Design

Inhalt

Grundlagen Game Design	3
Grafiken	3
Speicher	3
Ereignisse	3
Schleife	3
Grundlagen Programmierung	4
Wie fängt man an?	4
Stück für Stück	4
Paarprogrammierung	4
Wie funktionieren die Übungen?	5
Speichern und Laden in Scratch	5
Einen Scratch Account anlegen	6
Grundlagen-Übungen	7
Einfache Bewegung	8
Herausforderung	10
Objekte fangen	11
Herausforderung	16
Eine Geschichte	17
Herausforderungen	23
Spielvorlagen	24
Rennen	25
Endless Runner	28
Labyrinth	31
Quiz	34
Jump n' Run	38
Top-Down Shooter	41
• Interaktive Geschichte	45
Adventure	46
Und weiter?	47

Game Design

Grundlagen Game Design

Ein Spiel besteht in der Regel immer aus den gleichen Komponenten, die ihr auch in Scratch findet. Die meisten der vier Punkte findet ihr auch in den Beispielen wieder. Achtet darauf wie sie verwendet werden, denn sie kommen nicht nur bei Scratch vor, sondern auch in vielen anderen Programmiersprachen.

Grafiken

Grafiken werden auch Sprites genannt. Dazu zählen der Hintergrund, Spielerfiguren, Gegner, Punktezähler und Texte. Sprites sind Objekte, die eine Größe, einen Namen, ein Aussehen, eine Position, eine Drehrichtung und ein Verhalten haben. In Scratch wird Sprite mit Figur übersetzt.

Speicher

Speicher werden in der Informatik Variablen genannt. Variablen sind Objekte, die sich was merken können. Variablen haben einen Namen und einen Wert, der in ihnen gespeichert ist. In Scratch findet man Variablen unter „Variablen“

Ereignisse

Mit Ereignissen kann das Verhalten eines Spiels oder eines Sprites gesteuert werden. Ein Ereignis ist der Start von einem Algorithmus, mit dem das Spiel verändert wird. Ein Ereignis ist zum Beispiel das Drücken einer Taste, das Klicken auf einen Sprit, das Klicken der grünen Flagge oder wenn sich was am Spiel ändert. In Scratch findet man Ereignisse unter „Ereignisse“

Schleife

Ganz viele Abläufe in einem Spiel werden einmal gestartet und laufen dann so lange, bis etwas Bestimmtes passiert oder bis das Spiel beendet wird. Das kann zum Beispiel der Check sein, ob etwas berührt wird oder eine dauerhafte Bewegung. In Scratch findet man Schleifen unter „Steuerung“

Game Design

Grundlagen Programmierung

Damit das Spiel auch was wird und man nicht immer viel ändern muss gibt es ein paar Tipps, die auch von Profis benutzt werden, wenn sie Spiele (oder auch Apps) erstellen.

Wie fängt man an?

Am besten fangt ihr mit den Elementen des Spiels an die sich nicht bewegen: Hintergrund, Figuren, Schriften usw. So bekommt ihr erstmal ein Gefühl für das Aussehen und was das Spiel machen soll und was möglich ist.

Als zweites fangt ihr dann an euch die Ereignisse zu überlegen und diese Stück für Stück zu programmieren. WICHTIG: Immer alles testen. Es ist nichts ärgerlicher als viel zu bauen, was am Ende nicht funktioniert.

Stück für Stück

Wichtig ist, dass ihr nicht versucht alles auf einmal zu machen. Beim Programmieren ist es wichtig zu lernen, dass man erst ein Problem nach dem anderen löst und nicht alles auf einmal. Fang mit der einfachsten Aufgabe an und macht erst danach die schwereren Aufgaben.

Paarprogrammierung

Paarprogrammierung ist eine übliche Möglichkeit wie man zu zweit programmiert. Dabei gibt es zwei Rollen für die beiden Programmierer. Die Rollen werden regelmäßig gewechselt, so dass jeder mal jede Rolle innehat. Die beiden Rollen die es gibt sind: Fahrer und Navigator. Der **Fahrer** hat die Kontrolle über die Maus und die Tastatur und programmiert das Programm. Der **Navigator** unterstützt den Fahrer und überprüft, ob dieser Fehler macht.



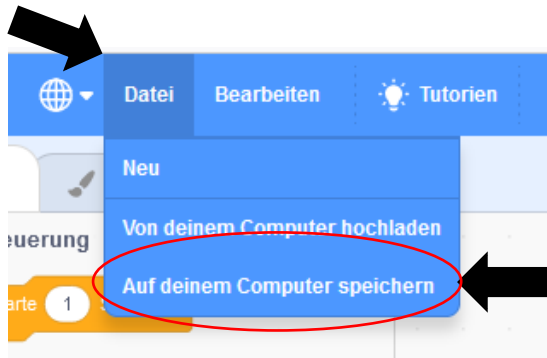
Game Design

Wie funktionieren die Übungen?

Eure Aufgabe ist es als erstes das Spiel „Einfache Bewegung“ nachzubauen und dann die Herausforderungen zu machen. Danach macht ihr das Gleiche mit „Objekte fangen“ und „Eine Geschichte“.

Damit solltet ihr die wichtigsten Grundlagen haben. Der nächste Schritt ist dann, dass ihr euch eine Vorlage, die eurem Spiel am ähnlichsten ist, raussucht. Baut die Vorlage nach und versucht sie dann eurem Spiel anzupassen. Ihr könnt auch gerne versuchen mehrere Vorlagen zu kombinieren.

Speichern und Laden in Scratch

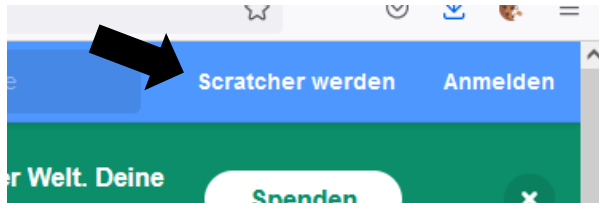


Ganz wichtig ist immer das Speichern! Viele der Aufgaben dauern länger als 40 Minuten. Damit du beim nächsten Mal weiter machen kannst musst du dein Spiel speichern und wieder laden.

Game Design

Einen Scratch Account anlegen

Eine Möglichkeit nicht zu speichern und seine Spiele zu teilen ist einen Account bei Scratch zu erstellen. Dann werden die Spiele alle immer gespeichert, es können keine Daten verloren gehen und ihr könnt eure Entwürfe auch mit anderen teilen und veröffentlichen.

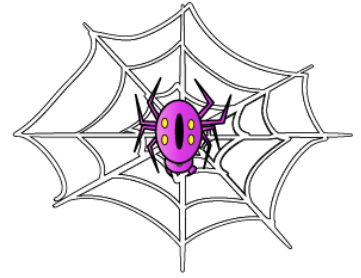


Um einen Account zu erstellen, geht man als erstes auf die Scratch Seite

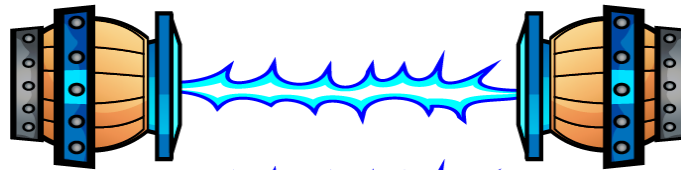
(<https://scratch.mit.edu>) und dann klickt man auf **Scratcher werden**.

A screenshot of the 'Scratcher werden' registration form. The form is titled 'Scratcher werden' and includes the text 'Erstelle Projekte, teile Ideen, finde Freunde. Es ist kostenlos!'. It has two main sections: 'Wähle einen Benutzernamen' with a text input field labeled 'Benutzername', and 'Passwort erstellen' with two text input fields labeled 'Passwort' and 'Gib dein Passwort erneut ein'. There is a checkbox labeled 'Passwort anzeigen' which is checked. At the bottom of the form is a large orange button labeled 'Weiter'.

Im zweiten Schritt gebe einen Benutzernamen und ein Passwort ein. Danach geht es weiter mit **Weiter**.



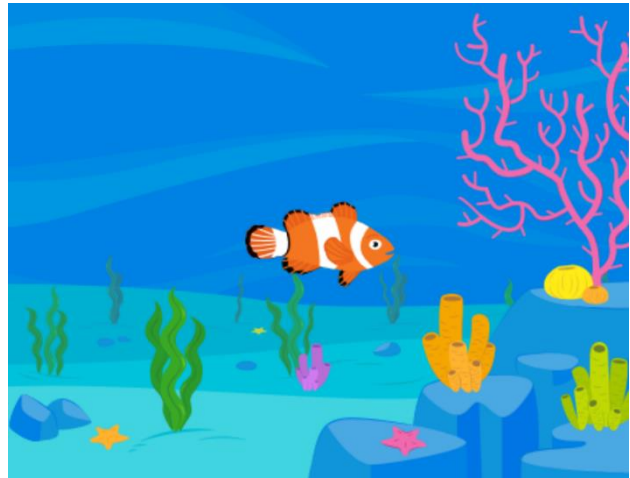
Grundlagen-Übungen



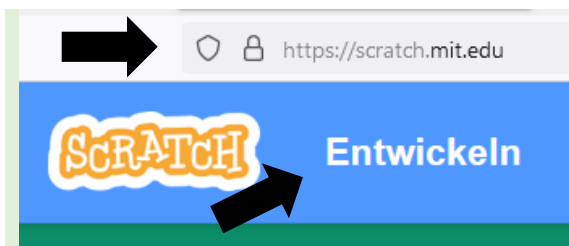
Game Design

Einfache Bewegung

Als erste Übung werdet ihr ein kleines Aquarium programmieren, bei dem ihr den Fisch mit den Pfeiltasten steuern könnt.

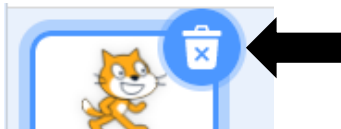


Aktion



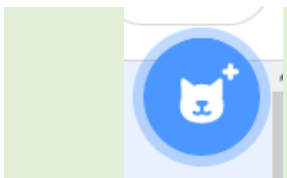
Erklärung

Gehe als erstes auf die Scratch-Seite. Öffne dafür einen Browser (Chrome, Firefox oder Edge) und gebe in die Adressleiste **scratch.mit.edu** ein. Gehe dann auch **Entwickeln**.

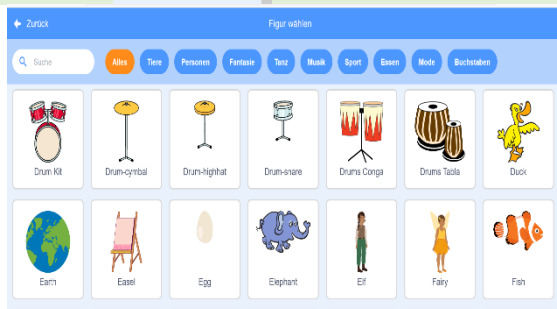


Für unser Aquarium ist die Katzen-Figur nicht passend, deswegen ersetzen wir sie. Dafür muss sie als erstes gelöscht werden.

Lösche die Figur mit dem „Mülleimer“-Knopf.

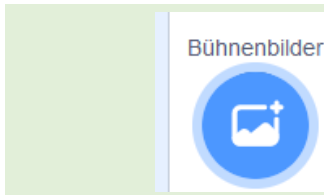


Jetzt fügen für eine neue Figur hinzu. Drücke dafür den „Bären“-Knopf.

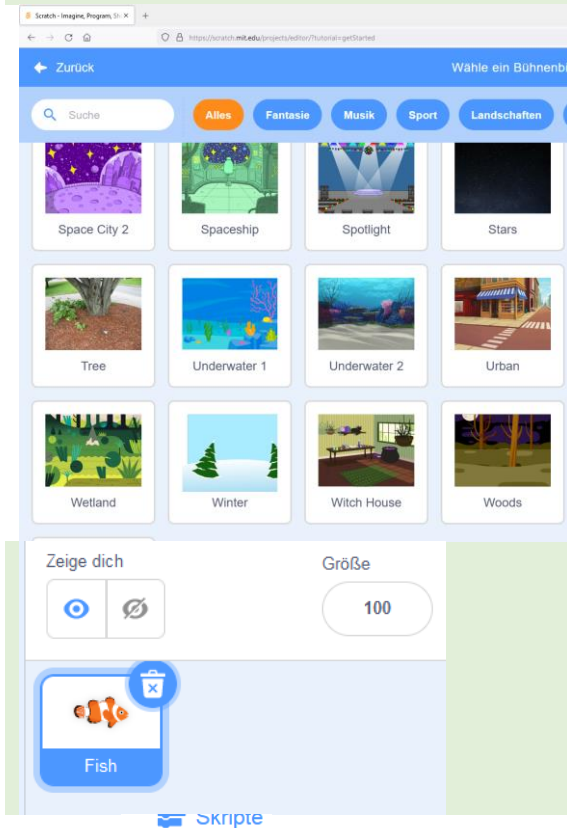


Suche jetzt den Fisch (oder ein anderes passendes Meerestier) und wähle es aus.

Game Design



Jetzt brauchen wir noch einen passenden Hintergrund. Klicke dafür auf den „Foto“-Knopf, neben dem Knopf für die Figuren.



Jetzt wähle ein passendes Hintergrundbild für dein Aquarium aus.

Jetzt programmieren wir den Fisch so, dass er sich bewegt. Klicke dafür den Fisch in der Figuren-Übersicht an.



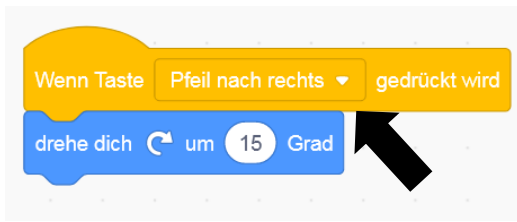
Alle Befehle zum Programmieren des Aquariums findest du ganz links bei **Bewegung**, **Ereignisse** und **Steuerung**.



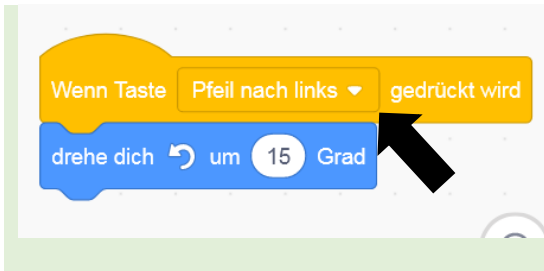
Füge jetzt die Blöcke aus dem linken Menü, wie auf dem Bild, in der Mitte des Bildschirms zusammen. Die Farben geben an, wo du die Blöcke findest.

Ändere die **10** in eine **2**.

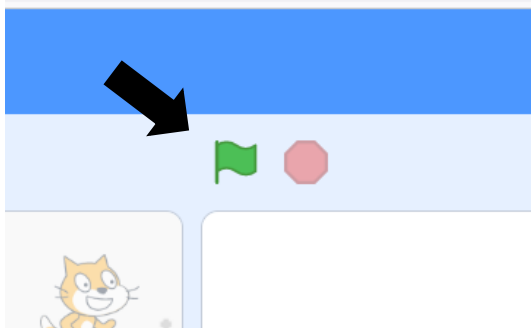
Game Design



Jetzt wird die Steuerung programmiert. Suche dafür den Block **Wenn Taste Leertaste gedrückt wird** und ziehe ihn rüber. Ändere dann **Leertaste** in **Pfeil nach rechts**. Füge danach einen **drehe dich** Block hinzu.



Mache jetzt das gleiche für die Drehung in die andere Richtung.



Dein Aquarium ist jetzt fertig. Du kannst es ausprobieren, indem du auf die grüne Fahne klickst. Der Fisch fängt dann an sich zu bewegen. Du steuerst ihn mit den **Pfeil-nach-links** und **Pfeil-nach-rechts**.

Herausforderung



Füge eine zweite Figur hinzu. Die Figur soll ein Taucher sein. Der Taucher soll mit den Tasten **a** und **d** ähnlich wie der Fisch gesteuert werden.



Erstelle ein neues Spiel mit einem Papagei und Dschungel. Der Papagei soll sich nach rechts und links bewegen, wenn die Pfeiltasten gedrückt werden und ansonsten sich nicht bewegen.

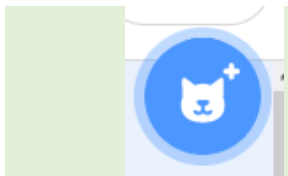
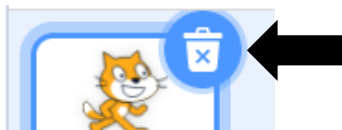
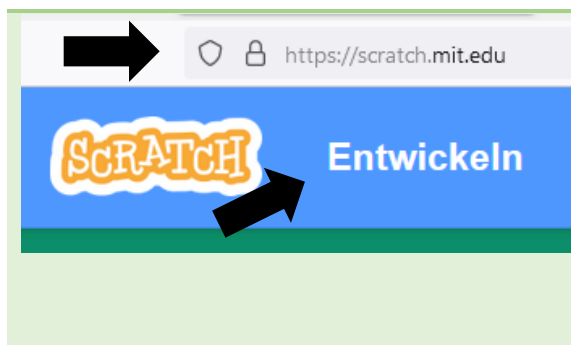
Game Design

Objekte fangen

In unserem zweiten Projekt folgt die Figur der Mausbewegung und interagiert mit anderen Figuren. Unser Raumfahrer soll die Kristalle sammeln, die sich bewegen und bei Berührung verschwinden.



Aktion



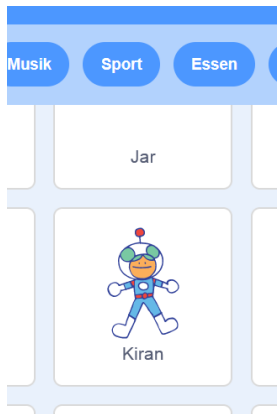
Erklärung

Die ersten Schritte unseres Programms sind die gleichen wie bei dem ersten Programm. Gehe als erstes auf die Scratch-Seite. Öffne dafür einen Browser (Chrome, Firefox oder Edge) und gebe in die Adressleiste **scratch.mit.edu** ein. Gehe dann auch **Entwickeln**.

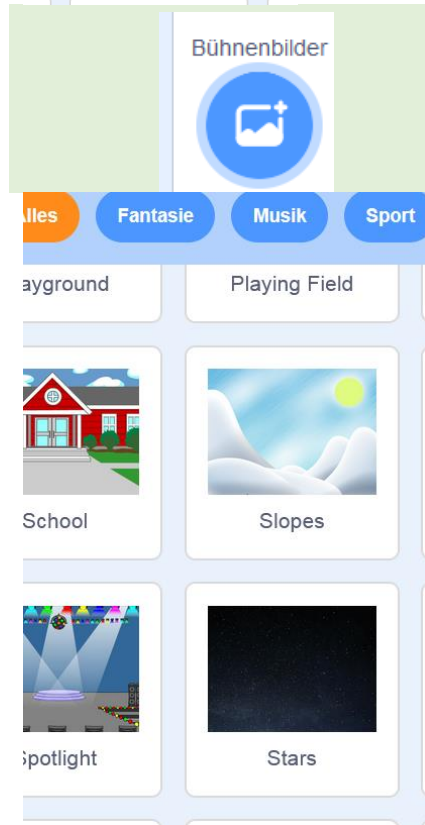
Für unser Spiel ist die Katzen-Figur nicht passend, deswegen ersetzen wir sie. Dafür muss sie als erstes gelöscht werden. Lösche die Figur mit dem „Mülleimer“-Knopf.

Jetzt fügen für eine neue Figur hinzu. Drücke dafür den „Bären“-Knopf.

Game Design



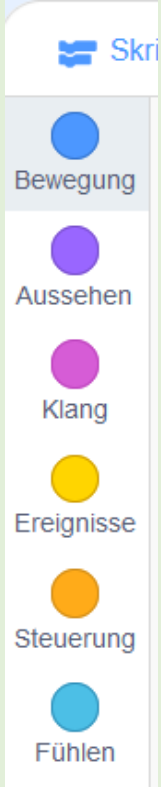
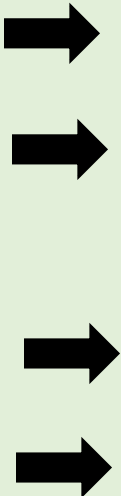
Suche jetzt Kiran und wähle ihn aus.



Jetzt brauchen wir noch einen passenden Hintergrund. Klicke dafür auf den „Foto“-Knopf, neben dem Knopf für die Figuren.

Jetzt wähle ein passendes Hintergrundbild für unser Spiel. Am besten den **Stars** Hintergrund.

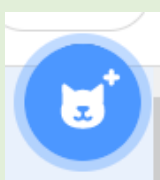
Game Design



Für die Programmierung unseres Raumfahrers brauchen wir die gezeigten Programmgruppen.



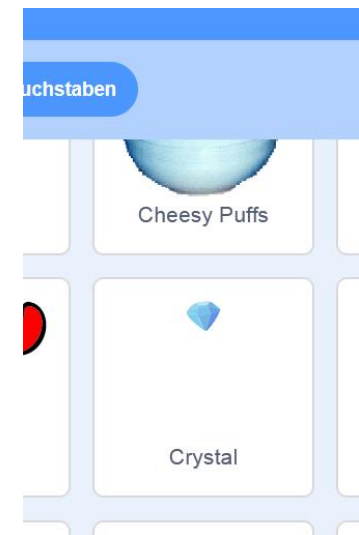
Programmiere nun Kiran den Raumfahrer mit den gezeigten Programmblöcken.



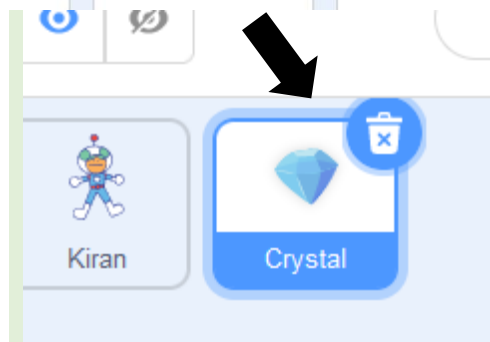
Als nächstes brauchen wir die Kristalle. Dafür fügen wir als ersten **eine** neue Figur hinzu.

Dafür wird wieder der „Bären“-Knopf gedrückt.

Game Design



Wähle jetzt den Kristall aus der Liste der verfügbaren Bilder.

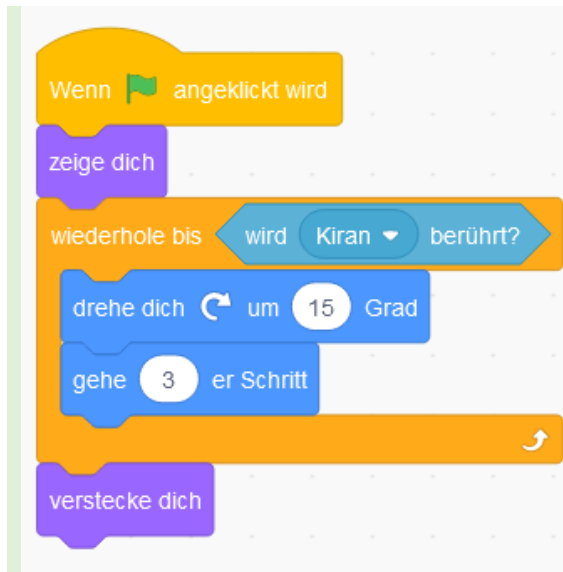


Wähle nun in der Figuren Übersicht den Kristall aus.

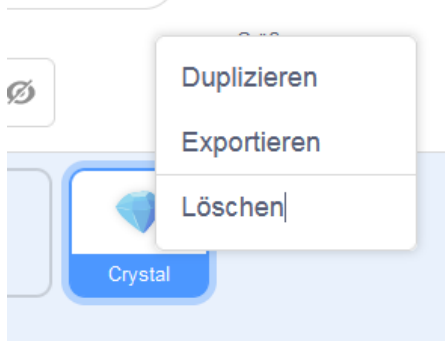


Die Blöcke für die Kristalle finden sich in den folgenden Gruppen.

Game Design

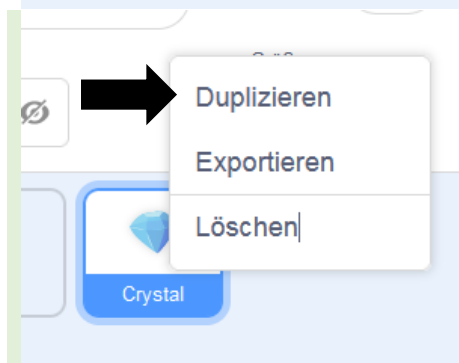


Suche jetzt die passenden Blöcke und erstelle den Code für den Kristall.



Jetzt haben wir unseren Raumfahrer und einen Kristall. Wir wollen aber mehr als einen Kristall haben.

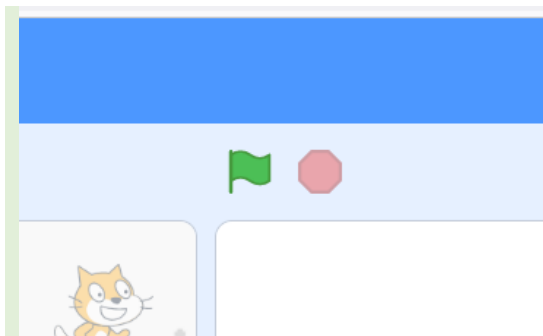
Dafür müssen wir den Kristall duplizieren. Gehe dafür mit dem Mauszeiger über den Kristall und klicke mit der rechten Maustaste.



Klicke jetzt auf **Duplizieren**.

Führe die oberen zwei Schritte noch ein paar Mal aus, bis du 10 Kristalle hast.

Jetzt kannst du das Spiel testen.



Game Design

Herausforderung



Baue das Spiel so um, dass eine andere Figur Äpfel in einem Zimmer aufsammelt. Die Äpfel sollen sich nicht im Zimmer bewegen, aber bei jedem Spielstart an einer zufälligen Position auftauchen.



Modifiziere das Apfelspiel so, dass es jetzt noch eine zweite Sorte an Gegenstände gibt. Die Süßigkeiten sollen nicht eingesammelt werden. Wenn die Spielfigur sie berührt, dann soll das Spiel vorbei sein.

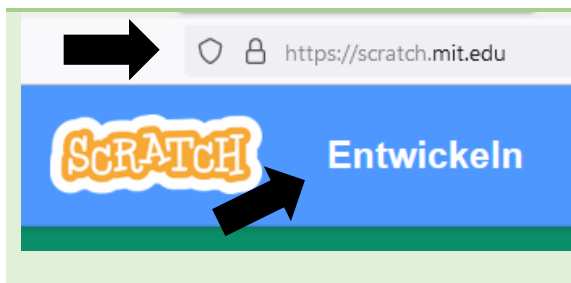
Game Design

Eine Geschichte

Bei diesem Spiel erzählen wir eine Geschichte mit Figuren und ändern die Szene. Es ist nur eine sehr kurze und keine interaktive Geschichte. Die findet ihr weiter hinten.

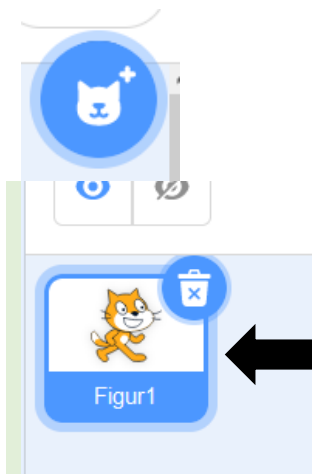


Aktion



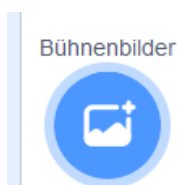
Erklärung

Erstellt als erstes ein neues Projekt in Scratch. Wie das geht wisst ihr ja inzwischen.



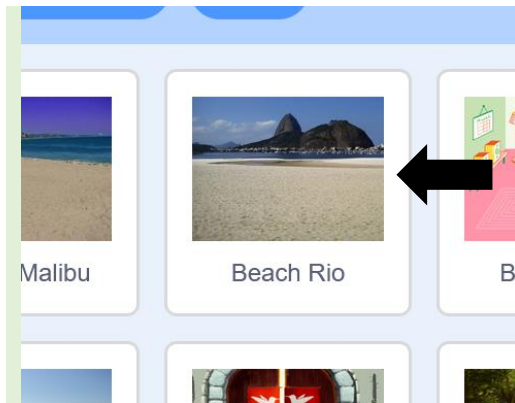
Als erstes brauchen wir eine zweite Figur für unsere Geschichte. Suche dir eine beliebige Figur aus und füge sie hinzu.

Jetzt wollen wir die Katze was sagen lassen. Klicke dafür auf die Katze.



Wir brauchen noch einen passenden Hintergrund für unser Spiel. Klicke auf den Knopf für die Hintergründe.

Game Design



Wähle **Beach Rio**.

Gebe der Katze jetzt den folgenden Code.

```
Wenn  angeklickt wird
  sage Hallo! für 2 Sekunden
  sage Ich bin Scartch. für 2 Sekunden
  sage Wer bist du? für 2 Sekunden
```

```
Wenn  angeklickt wird
  sage Hallo! für 2 Sekunden
  sage Ich bin Scartch. für 2 Sekunden
  sage Wer bist du? für 2 Sekunden
  sende Nachricht1 an alle
```

```
sende Nachricht1 an alle
```

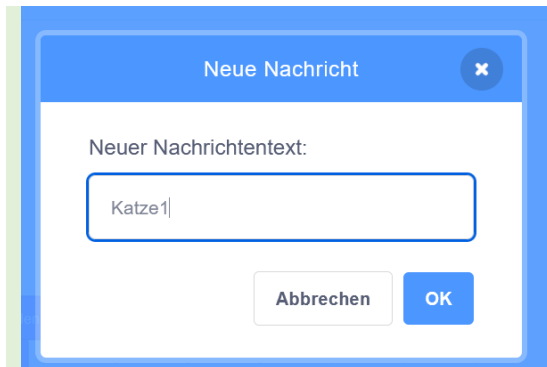
Neue Nachricht

✓ Nachricht1

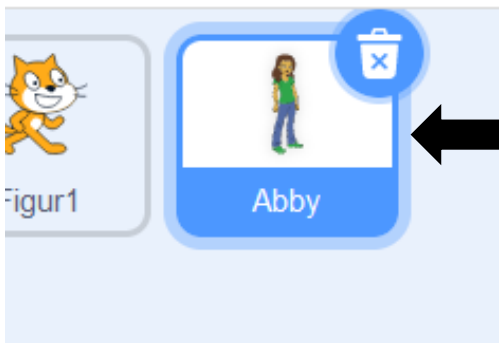
Damit die zweite Figur der Katze antworten kann, müssen wir eine Nachricht verschicken, damit die Figur weiß, dass die Katze fertig ist. Füge deswegen noch folgenden Code zu der Katze hinzu.

Klicke jetzt auf **Nachricht1** und dann auf **Neue Nachricht**, um eine neue Nachricht zu erstellen.

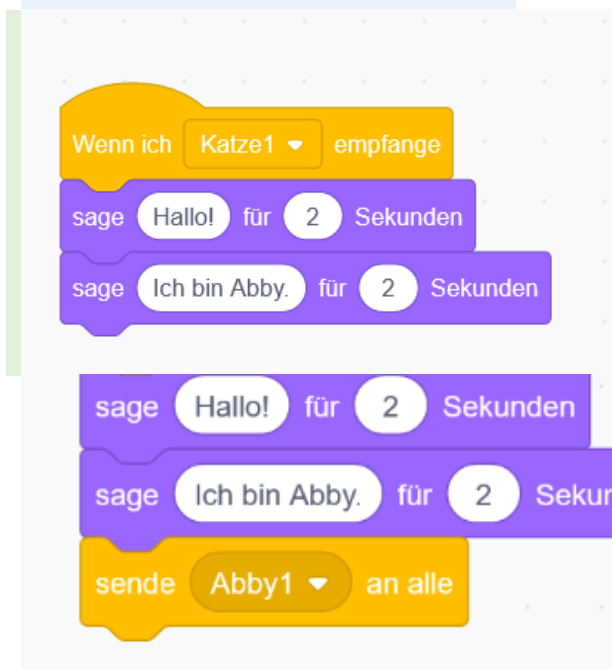
Game Design



Gebe der Nachricht den Namen **Katze1**.

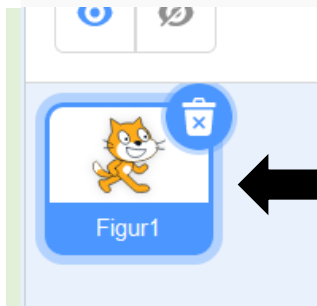


Wähle jetzt die zweite Figur aus.



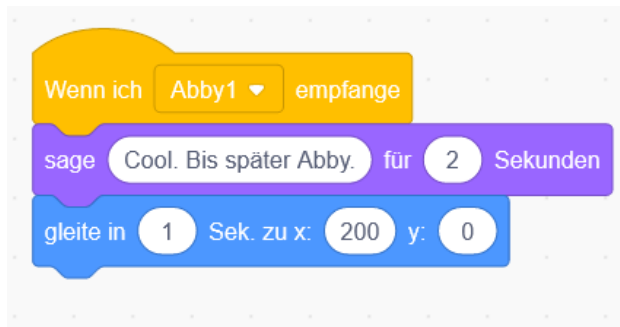
Gebe der Figur jetzt die Blöcke für die Antwort.

Damir die Katze weiß, wann Abby fertig ist brauchen wir eine weitere Nachricht. Erstelle eine neue Nachricht und nenne diese **Abby1**



Jetzt lassen wir die Katze auf die Nachricht **Abby1** reagieren. Wechsel dazu wieder zur Katze.

Game Design

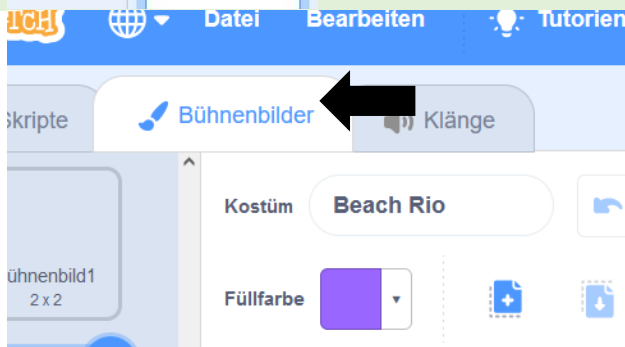


Jetzt geben wir der Katze weiteren Code, damit zu zum Rand von dem Bildschirm läuft.

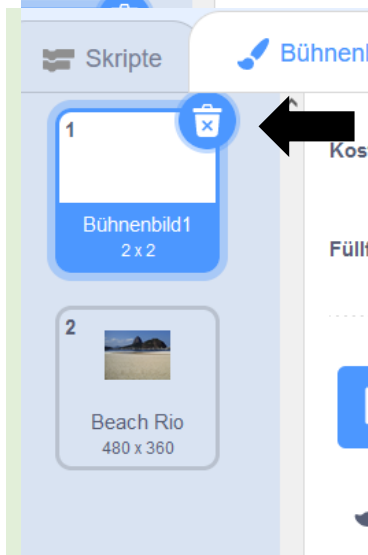


Wenn unsere Figur am Rand ankommt, dann wollen wir den Hintergrund wechseln und Abby soll verschwinden. Dafür brauchen wir erst einen neuen Hintergrund.

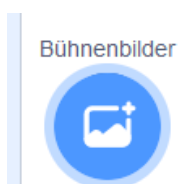
Wähle dafür das Bühnenbild aus.



Jetzt gehe auf Bühnenbilder.

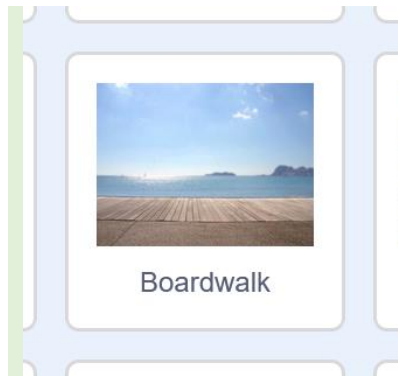


Klicke auf **Bühnenbild1** und dann auf den Mülleimer, um den leeren Hintergrund zu löschen.



Klicke auf den Hintergrund-Knopf, um den Hintergrund **Boardwalk** hinzuzufügen.

Game Design



Wähle den Hintergrund
Boardwalk.



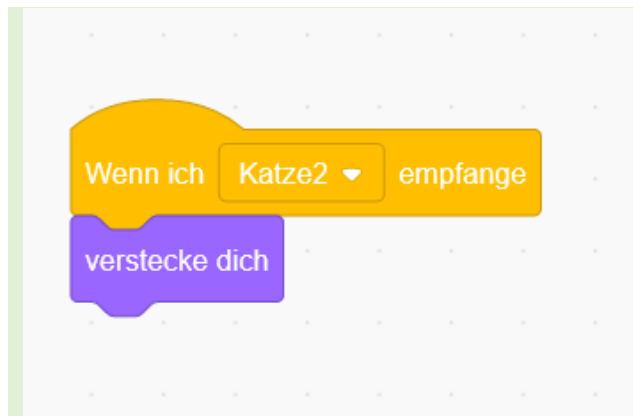
Jetzt müssen wir der Katze sagen, welche Hintergrund beim Start erscheinen soll. Dafür müssen wir einen Block bei der Katze einfügen.

Und das gleiche machen wir, wenn die Katze am Rand angekommen ist.

Jetzt müssen wir Abby mitteilen, dass sie unsichtbar werden soll und die Katze müssen wir vom Rand bewegen.

Als erstes geben wir der Katze eine neue Nachricht: **Katze2**

Game Design



Wenn ich Katze2 empfangen

verstecke dich

Und Abby bekommt diese Blöcke.

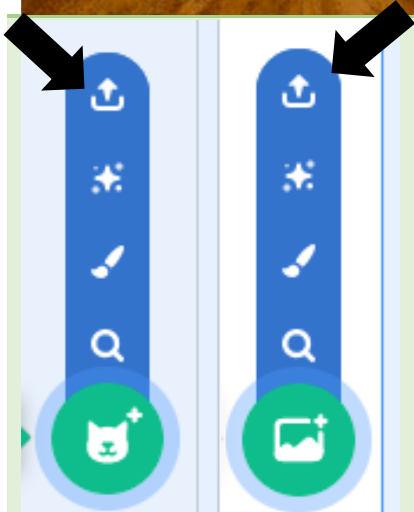
Jetzt könnt ihr die Geschichte weiterschreiben.

Game Design

Herausforderungen



Füge der Geschichte mehr Handlung hinzu. Es soll mindestens eine weitere Szene sein.



Erstelle eine neue Geschichte. Lasse deiner Kreativität freien Lauf. Scratch bietet die Möglichkeit, eigene Grafiken für Figuren und Hintergründe zu einem Spiel hinzuzufügen. Versuche die Geschichte mit ein paar eigene Bilder zu gestalten.



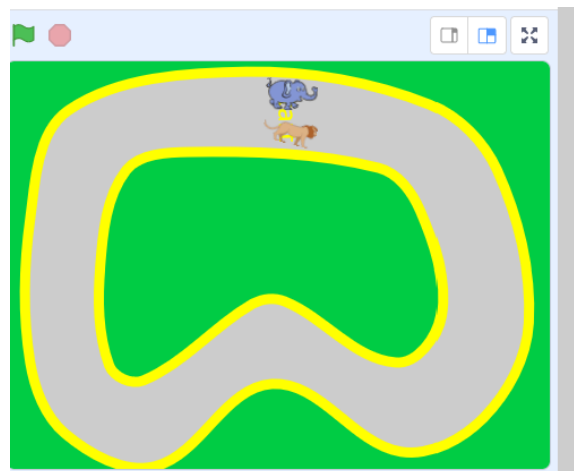
Spielvorlagen



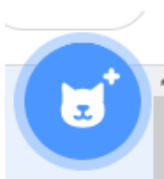
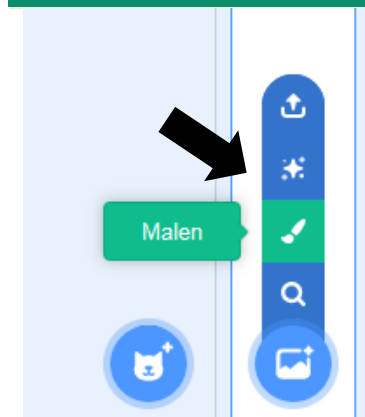
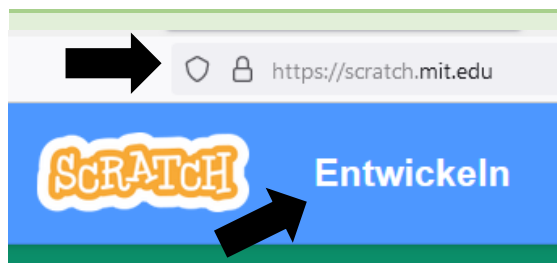
Game Design

Rennen

Bei diesem Spiel zeichnen wir eine eigene Rennstrecke und programmieren dann zwei Autos so, dass sie auf der Strecke um die Wette fahren. Das Spiel kann man zu zweit spielen. Wenn ihr die Strecke verlasst, werdet ihr langsamer.



Aktion



Erklärung

Erstellt als erstes ein neues Projekt in Scratch. Wie das geht wisst ihr ja inzwischen.

Als erstes zeichnen wir die Rennstrecke als Hintergrundbild. Gehe dafür mit der Maus das das Hintergrundsymbol und wähle Malen. **Nicht auf das Hintergrundsymbol klicken.**

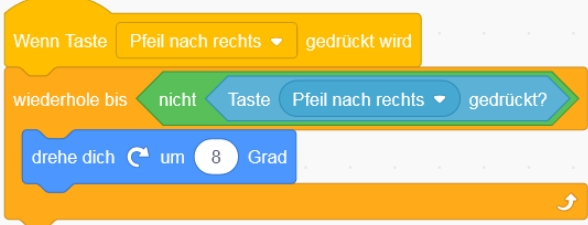
Deine Rennstrecke kann zum Beispiel so aussehen.

Füge jetzt zwei passende Figuren hinzu. Das könne Tiere oder Autos sein. Platziere die Figuren am Start.

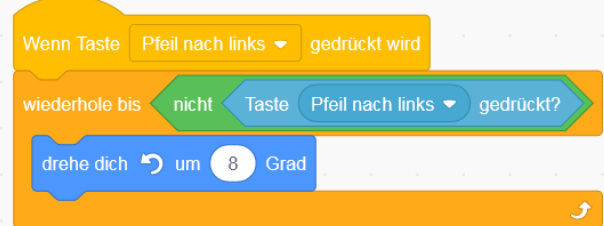
Game Design



Lösche jetzt die Katzen Figur.



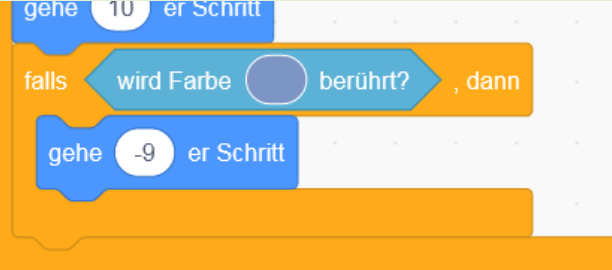
Wähle jetzt eine Figur aus und füge Blöcke für recht drehen hinzu.



Füge jetzt einen Block für nach links drehen hinzu.



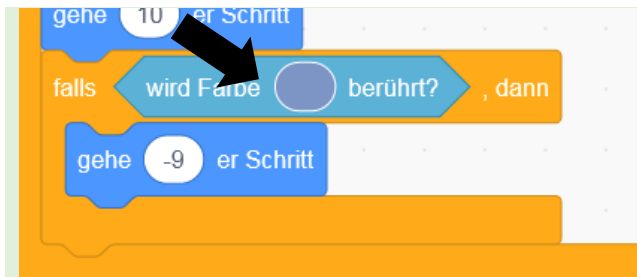
Jetzt fehlt noch die Bewegung gerade aus.



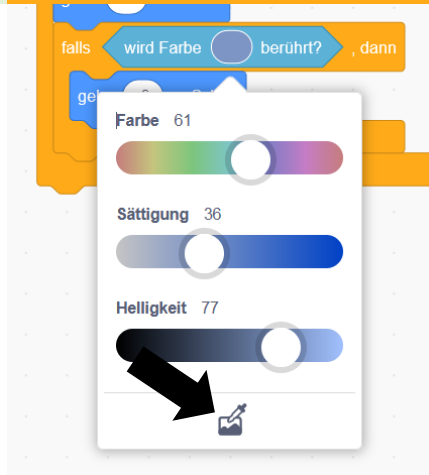
Wähle jetzt die zweite Figur, füge die gleichen Blöcke hinzu und ändere die verwendeten Tasten. **w** für geradeaus, **d** für nach rechts und **a** für nach links

Jetzt wollen wir das Spiel so ändern, dass sich die Figuren auf dem Rasen langsamer bewegen. Füge dafür die folgenden Blöcke nach **gehe 10er Schritt** ein

Game Design



Noch ändert sich nichts. Wir müssen die Farbe in dem Block anpassen. Klicke dafür auf die Farbe.

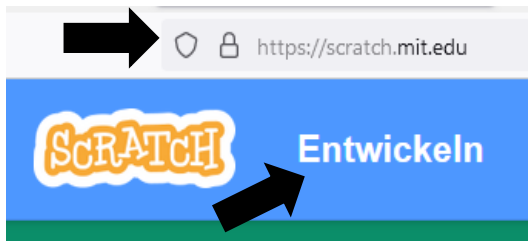


Verwende jetzt das Pipettenwerkzeug, um die passende Farbe im Hintergrund zu wählen.

Game Design

Endless Runner

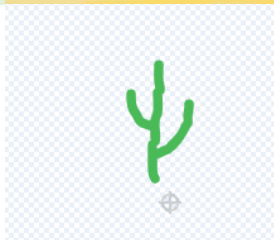
Bei diesem Spiel läuft eure Figur ohne Pause und es erscheinen immer neue Hindernisse, die ihr überspringen müsst. Es ist gar nicht so einfach. Viel Glück!



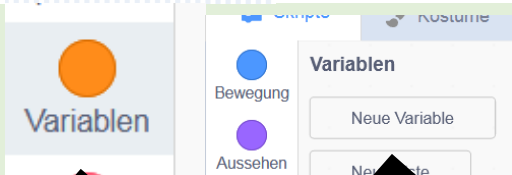
Erstellt als erstes ein neues Projekt in Scratch. Wie das geht, wisst ihr ja inzwischen.



Jetzt brauchen wir wie immer einen passenden Hintergrund. Wähle den Hintergrund „Desert“.

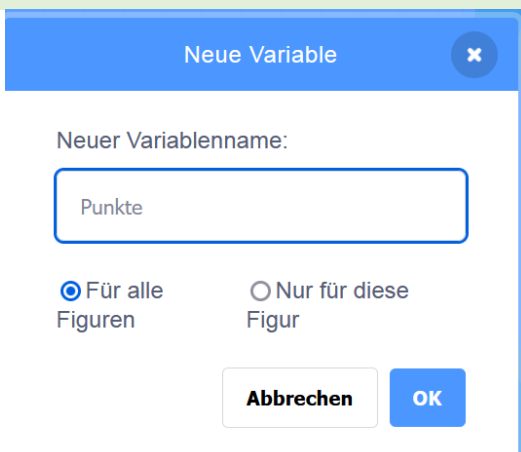


Als nächstes brauchen wir Hindernisse für unser Spiel. Dafür zeichnen wir eine neue Figur, die aussieht wie ein Kaktus.

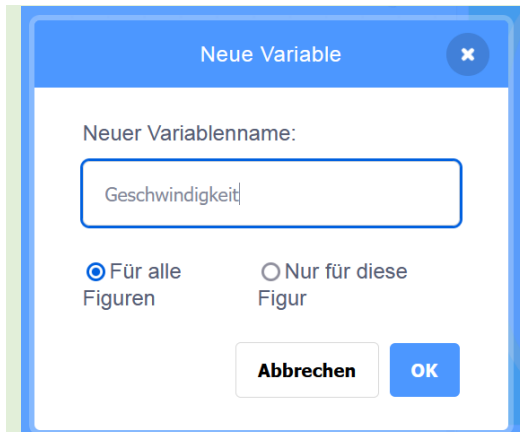


Wie bei dem Quiz brauchen wir auch hier eine Variable, um die Punkte zu zählen. Dafür fügen wir als erstes eine neue Variable hinzu.

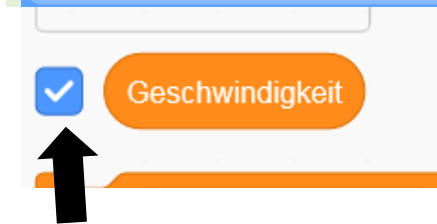
Wähle dafür das Menü **Variablen** und dann **Neue Variable**. Nenne die neue Variable **Punkte**.



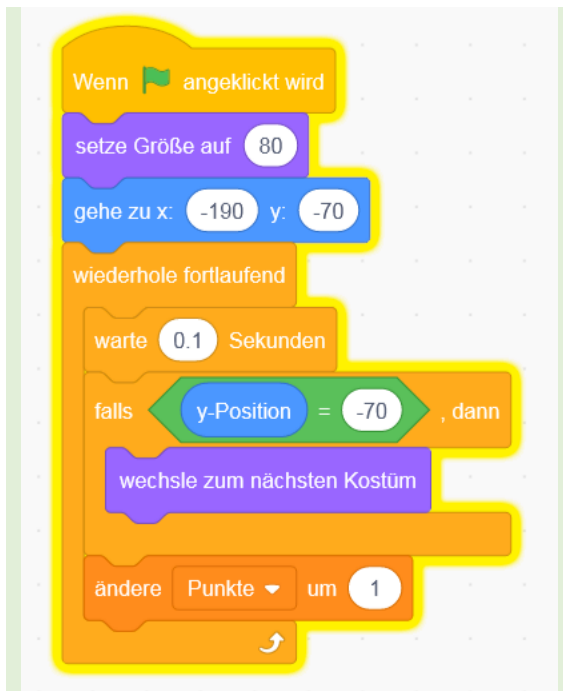
Game Design



Erstelle eine zweite Variable und nenne diese **Geschwindigkeit**. Diese Variable wird dafür benötigt, damit die Sprünge gut aussehen.



Die Variable Geschwindigkeit wollen wir verstecken, da wir sie nur für interne Berechnungen brauche. Klicke dafür auf das blaue Kästchen neben der Variablen.

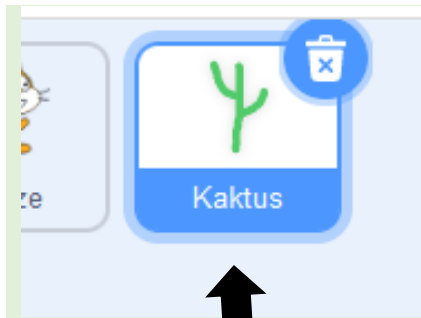


Programmiere jetzt die Katze, indem du folgende Blöcke zu der Katze hinzufügst.



Jetzt wollen wir die Katze springen lassen. Gebe ihr dafür die folgenden Blöcke.

Game Design



Als nächstes muss der Kaktus programmiert werden. Dafür kopiert der Kaktus sich und erzeugt durch seine Bewegung die Illusion einer echten Bewegung. **Klicke dafür den Kaktus an**



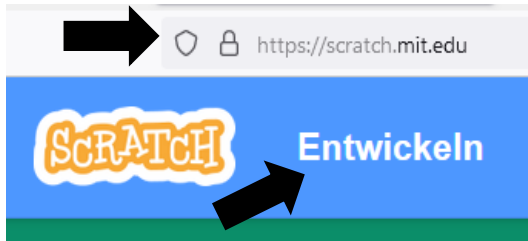
Jetzt wird programmiert, dass der Kaktus zufällig Klone von sich erzeugt, die sich auf den Spieler zubewegen.



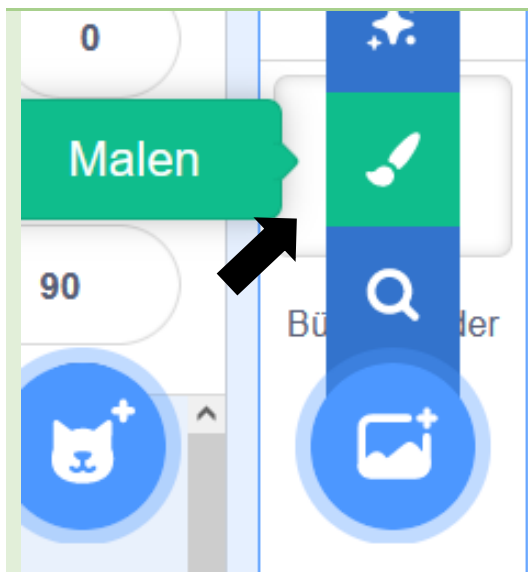
Game Design

Labyrinth

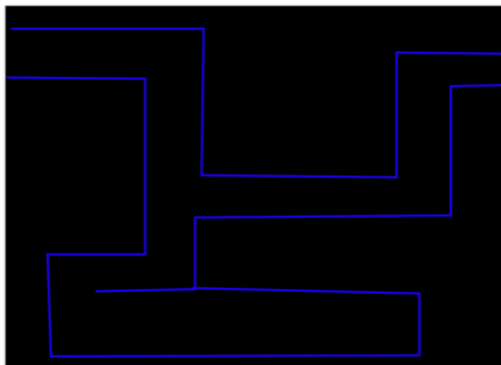
Bei diesem Spiel bewegt ihr eine Figur durch ein Labyrinth. Es ist ähnlich zu dem Rennspiel, nur dass eure Figur jetzt nicht durch die Wände gehen kann.



Erstellt als erstes ein neues Projekt in Scratch. Wie das geht, wisst ihr ja inzwischen.

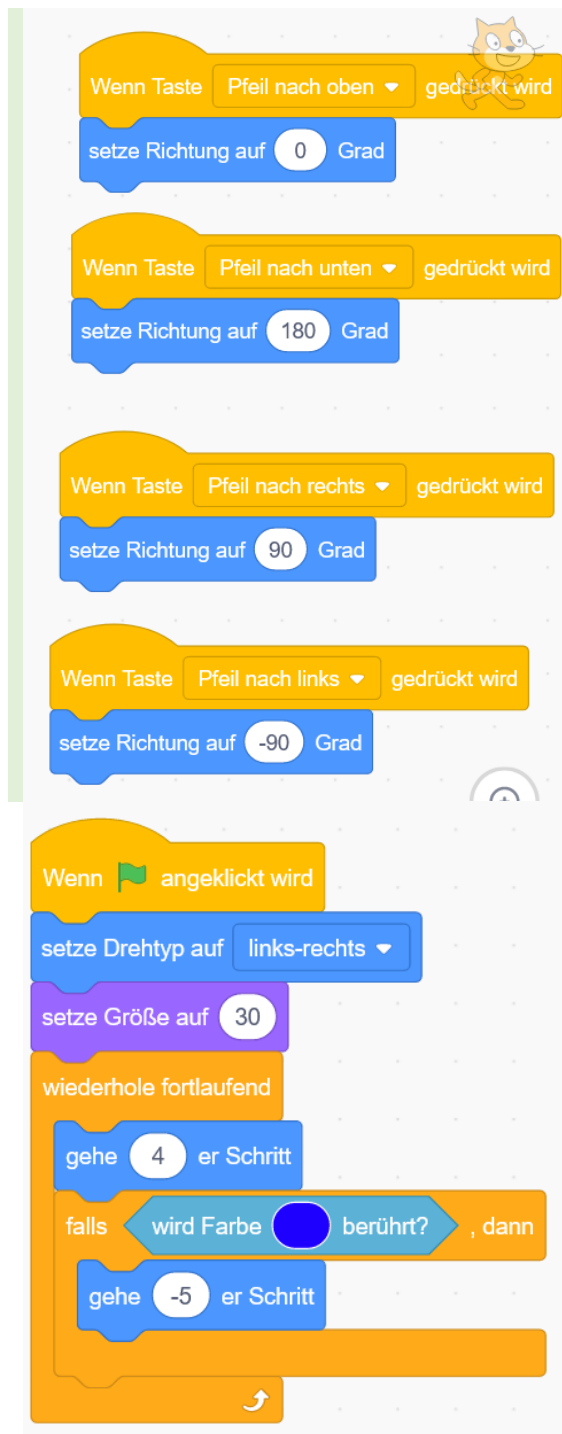


Als erstes muss ein Labyrinth als Hintergrund gezeichnet werden. Verwende dafür die Zeichenfunktion für einen neuen Hintergrund.



Färbe den Hintergrund schwarz und zeichne dann die Wände in einer andere Farbe.

Game Design



Nun wird die Steuerung der Figur programmiert.

Als letztes wird noch die Kollision mit den Wänden programmiert. Diese soll erst funktionieren, wenn die grüne Flagge geklickt wird.

Game Design



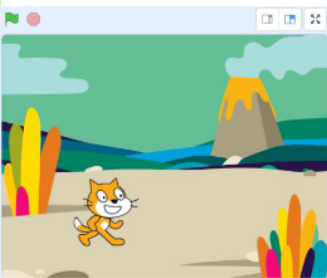
Ihr könnt auch eine alternative Bewegung programmieren direkt mit den Pfeiltasten und ohne automatische Bewegung.

Game Design

Quiz

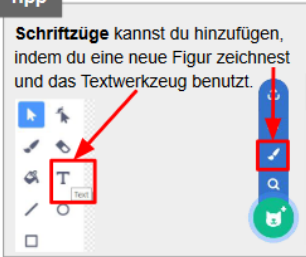
Jetzt werdet ihr ein Quiz programmieren. Das Programm stellt euch Fragen und ihr könnt dann eine Antwort eingeben. Für jede richtige Antwort gibt es einen Punkt.

4.1 Quiz: Start



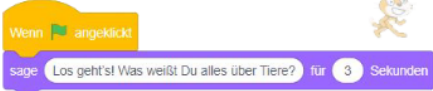
Du brauchst diese Komponenten:
1 Bühnenbild (z. B. "Jurassic")
1 Katze ("Cat1", "Figur1" oder andere Namen)

Tipp
Schriftzüge kannst du hinzufügen, indem du eine neue Figur zeichnest und das Textwerkzeug benutzt.



Aufgabe

- Gib scratch.mit.edu in deinen Browser ein und klicke auf "Entwickeln"
→ Dort kannst du programmieren.
Hinweis: Um deine Projekte abzuspeichern, musst du dich mit Benutzername und Passwort anmelden. Wie das funktioniert, erfährst du von deiner Lehrkraft.
- Ändere die **Sprache** bei Bedarf auf **Deutsch**.
- Links siehst du das Design des Spiels und die Komponenten, die du benötigst.
 - Wähle ein **Bühnenbild** aus der Bibliothek als Hintergrundbild aus.
 - Überlege dir ein **Thema für dein Quiz**, z. B. "Tiere". Füge deinen **Quiznamen** als Schriftzug hinzu.
- Lass die Katze **bei Spielstart sagen**: "Los geht's! Was weißt Du alles über [Thema]?"



Nächste Aufgabe

- Erstelle eine Variable "Richtige_Antworten".
- Erstelle selbst einen neuen Block und nenne diesen "Quiz_Teil1". Führe den Block aus, nachdem die Katze gesprochen hat.
Tipp: Eigene Blöcke kannst du im Bereich "Meine Blöcke" erstellen.
- Definiere deinen Block: Setze die Variable auf 0 und stelle eine erste Quizfrage.

www.appcamps.de

4.2 Quiz: Teil 1 definieren

APP CAMPS



Du brauchst diese Komponenten:



Tipp

Im Reiter "Skripte" kannst du unter "Meine Blöcke" neue, eigene Programmierblöcke erstellen, definieren und danach wiederverwenden.



Aufgabe

1. Erstelle eine **Variable** "Richtige_Antworten".
2. Erstelle selbst einen neuen **Block** und nenne diesen "Quiz_Teil1".
3. **Definiere** deinen Block:
 - a. Setze die **Variable** "Richtige_Antworten" auf 0.
 - b. **Stelle** deine erste **Quizfrage**, z. B. "Welcher Vogel legt seine Eier in fremde Nester?".

Hinweis: Nachdem eine Frage gestellt wurde, erscheint immer automatisch ein Eingabefeld für eine Antwort.
4. **Führe den Block** "Quiz_Teil1" **aus**, nachdem die Katze gesprochen hat.



Nächste Aufgabe

Überprüfe die eingegebene Antwort. Je nachdem, ob die Antwort richtig oder falsch ist, soll die Katze unterschiedlich reagieren. Benutze dazu eine bedingte Anweisung (falls...dann...sonst) und mathematische Gleichungen, die du unter "Operatoren" findest.

www.appcamps.de



4.3 Quiz: Antworten prüfen und weitere Fragen

APP CAMPS



Du brauchst diese Komponenten:



Tipp

Es gibt einen Trick, wie du weitere Fragen sehr einfach hinzufügen kannst: Klicke mit Rechtsklick auf den Programmierblock (hier: "falls...") und wähle **duplizieren**. Jetzt musst du nur noch die Fragen und Antworten anpassen.

Aufgabe

1. Überprüfe die eingegebene Antwort. Je nachdem, ob die Antwort richtig oder falsch ist, soll die Katze unterschiedlich reagieren:
 - a. **Wenn** die eingegebene **Antwort richtig ist**, spiele einen **Klang** ab, lass die Katze "**Richtig!**" **sagen** und **ändere** die **Variable** "Richtige_Antworten" **um 1**.
 - b. **Sonst** (wenn die eingegebene Antwort falsch ist), spiele einen anderen **Klang** ab und lass die Katze **sagen**: "**Leider falsch!**".

Hinweis: Benutze eine bedingte Anweisung (falls...dann...sonst) und einen Gleich-Operator (=), um zu prüfen, ob die Antwort richtig ist.
2. Füge zwei weitere **Fragen** hinzu und überprüfe jeweils die eingegebenen Antworten.
3. **Teste** dein Quiz, indem Du es mit der grünen Flagge startest und stoppst.



Nächste Aufgabe

Das Quiz soll zwei Schwierigkeitslevel haben: Quiz Teil 1 und Teil 2. Erstelle dafür ein weiteren Block "Quiz_Teil2".

www.appcamps.de



4.4 Quiz: Teil 2 definieren

APP CAMPS



Du brauchst diese Komponenten:

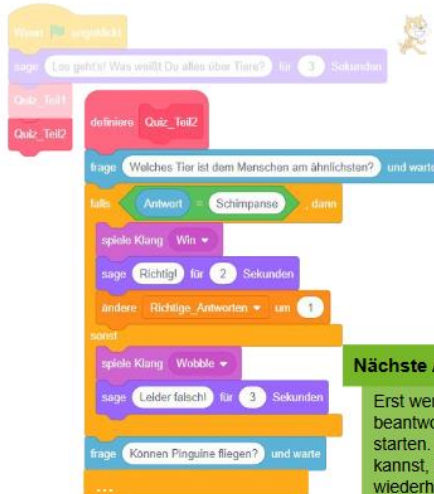


Hinweis

Klicke mit Rechtsklick auf den ersten "frage"-Block in Quiz_Teil1 und wähle **duplizieren**. Füge die Blöcke bei "definiere Quiz_Teil2" an und passe dann jeweils die Fragen und Antworten an.

Aufgabe

1. Das Quiz soll zwei Schwierigkeitslevel haben: Erstelle einen neuen Block **"Quiz_Teil2"** und **rufe ihn** nach Teil 1 **auf**.
2. Füge (wie im ersten Teil) **drei neue Fragen** hinzu und überprüfe danach die eingegebene Antworten.



Nächste Aufgabe

Erst wenn alle Fragen aus Teil 1 richtig beantwortet wurden, soll Teil 2 des Quiz starten. Überlege Dir, wie Du überprüfen kannst, ob der Spieler die Fragen aus Teil 1 wiederholen muss oder mit den Fragen aus Teil 2 weitermachen darf.

www.appcamps.de

4.5 Quiz: Fallunterscheidung

APP CAMPS



Du brauchst diese Komponenten:



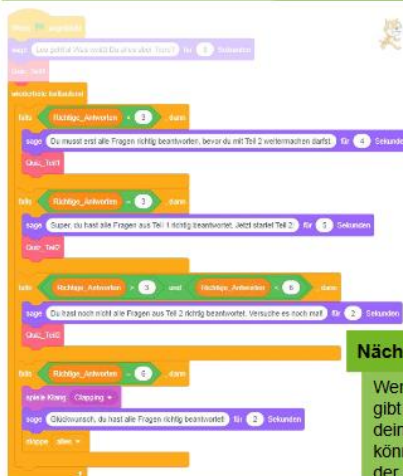
Tipp

Benutze mathematische Gleichungen unter "Operatoren", um zu prüfen, wie viele Antworten bereits richtig beantwortet wurden. Du kannst die Blöcke auch ineinander schachteln.



Aufgabe

1. Erst wenn alle Fragen aus Teil 1 des Quiz richtig beantwortet wurden, soll Teil 2 starten: **Überprüfe**, wie viele Antworten bereits **richtig beantwortet** wurden und **entscheide** entsprechend, ob der Spieler die Fragen aus **Teil 1** wiederholen muss oder mit den Fragen aus **Teil 2** weitermachen darf.
2. **Wenn** der Spieler alle Fragen korrekt beantwortet hat, **beglückwünsche** ihn.
3. **Teste** dein Quiz, indem du es mit der grünen Flagge startest. Gib auch mal falsche Antworten ein, um zu testen ob dein Quiz so funktioniert wie gedacht.



Nächste Aufgabe

Wenn der Spieler einen Fehler in Teil 2 macht, gibt es einen Fehler. Diskutiere zusammen mit deinem Sitznachbarn was der Fehler sein könnte und wie er behoben werden kann. Auf der nächsten Seite ist eine mögliche Lösung.

www.appcamps.de

4.6 Quiz: Fehler fixen

APP CAMPS



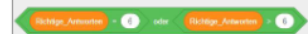
Du brauchst diese Komponenten:



Info

Beim Programmieren gibt es meistens mehr als eine richtige Lösung. Das ist auch hier der Fall.

Alternativ können wir zum Beispiel statt `Richtige_Antworten = 6` die Bedingung so ändern:



Aufgabe

1. **Setze die Variable "Richtige_Antworten" in Quiz_Teil 2 auf 3.**
Hinweis: Der Wert muss der Anzahl der Fragen entsprechen. Wenn du in den Quizteilen mehr als 3 Fragen gestellt hast, musst du das hier und bei den falls-dann-Anweisungen (Lernkarte 4.4) anpassen.
2. **Teste** dein Quiz, indem Du es mit der grünen Flagge startest.

definiere Quiz_Teil2



Erläuterung: Warum müssen wir die Variable `Richtige_Antworten` in Quiz_Teil2 auf den Wert = 3 setzen?

Wenn der Spieler in Teil 2 einen Fehler macht, hat er im ersten Versuch zum Beispiel 5 Punkte. Im zweiten Versuch wird die Variable wie im Code vorgegeben weiter hochgezählt, sodass der Spieler (wenn er alle Fragen korrekt beantwortet) 8 Punkte hat. Die Bedingung für die Beglückwünschung und das Ende des Quiz ist, dass die Variable = 6 ist. Die Bedingung trifft also nicht ein, wenn wir die Variable nicht zurücksetzen.

Nächste Aufgabe

- Erstelle eine neue Variable, die das aktuelle Level anzeigt.
- Ändere den Hintergrund, wenn der Spieler Teil 2 des Quiz erreicht.

www.appcamps.de

4.7 Quiz: Level unterscheiden

APP CAMPS



Du brauchst diese Komponenten:

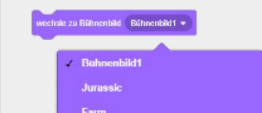


Tipp

Weitere Bühnenbilder kannst du zunächst über den Button **"Bühnenbild wählen"** hinzufügen.



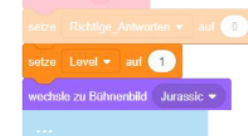
Anschließend kannst du die Bühnenbilder beim **"wechsle zu Bühnenbild"-Block** auswählen.



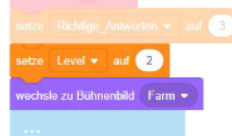
Aufgabe

1. Erstelle eine neue Variable **"Level"**, die zeigt, ob sich der Spieler im ersten oder zweiten Teil des Quiz befindet. **Setze die Variable** entsprechend dem jeweiligen Level auf 1 bzw. 2.
2. **Ändere das Bühnenbild**, wenn der Spieler Teil 2 des Quiz erreicht.
Hinweis: Definiere das Bühnenbild im Code sowohl für Teil 2 als auch für Teil 1.

definiere Quiz_Teil1



definiere Quiz_Teil2



Nächste Aufgabe

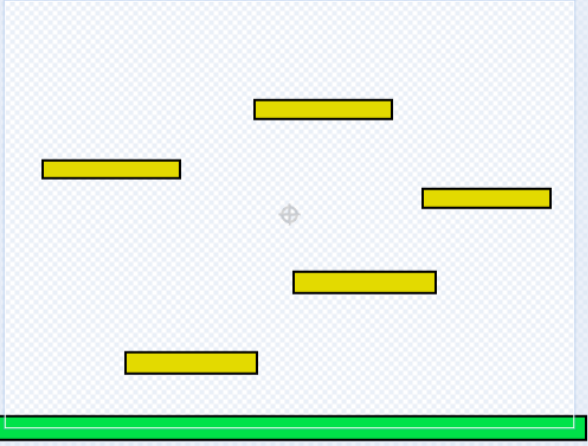
Hast du noch weitere Ideen, wie du das Quiz erweitern kannst? Dann kannst du das jetzt ausprobieren. Viel Spaß dabei!

www.appcamps.de

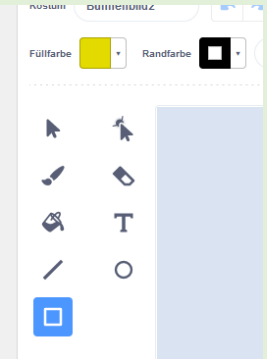
Game Design

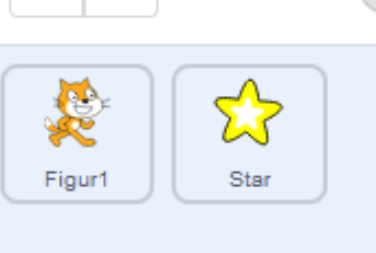
Jump n' Run

Jetzt kommt ein klassisches Spiel, bei dem ihr über Hindernisse springen müsst und zu einem Ziel kommen müsst.

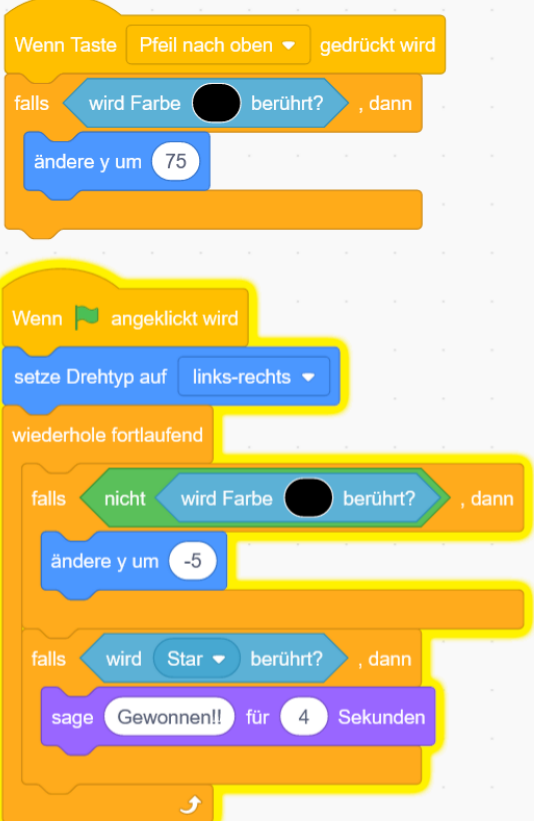


Malt einen Level als Hintergrund. Ihr könnt auch Figuren dafür benutzen, wenn ihr wollt. Damit das Beispiel funktioniert, brauchen eure Plattformen und der Boden eine schwarze Umrandung.





Erstellt dann zwei Figuren. Der Stern ist das Ziel.

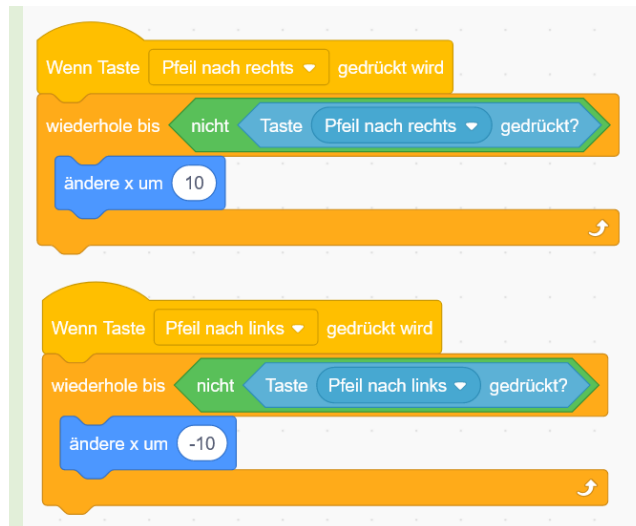


```
Wenn Taste Pfeil nach oben gedrückt wird
falls wird Farbe [schwarz] berührt? , dann
  ändere y um 75

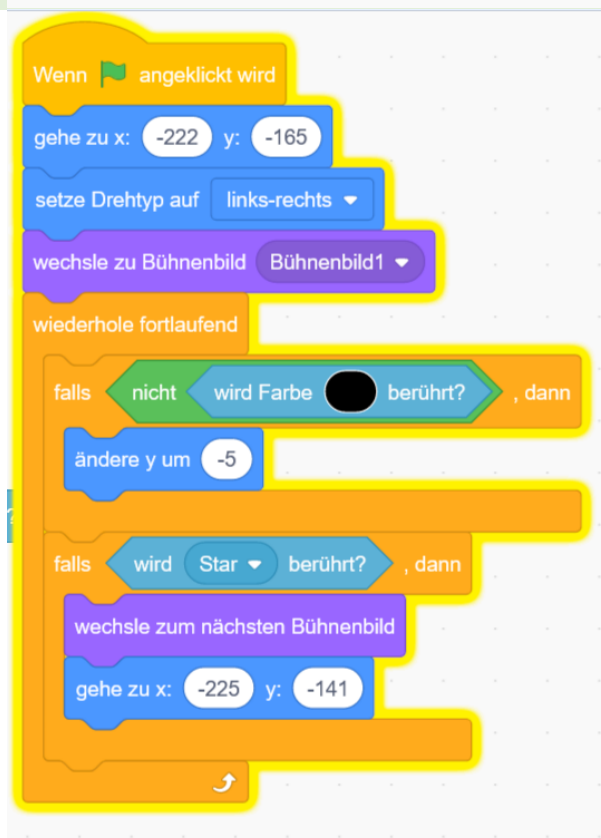
Wenn [grüner Flagge] angeklickt wird
  setze Drehtyp auf links-rechts
  wiederhole fortlaufend
    falls nicht wird Farbe [schwarz] berührt? , dann
      ändere y um -5
    falls wird Star berührt? , dann
      sage Gewonnen!! für 4 Sekunden
```

Hier ist der Code für die Katze.

Game Design

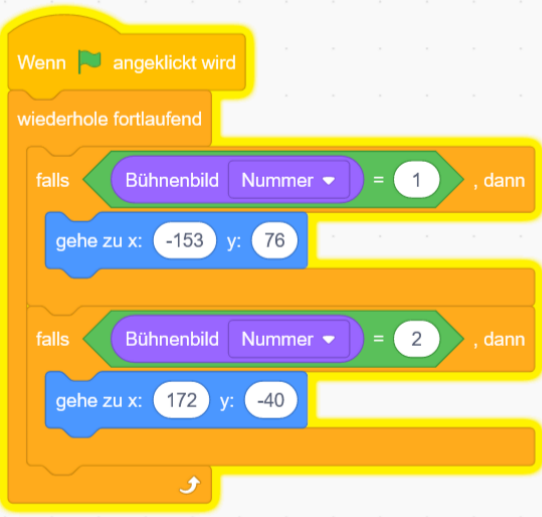


Hier ist noch mehr Code für die Katze.



Der Code auf der linken Seite ist eine Variante des Katzen Programms. Ihr könnt ihn verwenden, wenn ihr mehrere Level habt. Der Level wird gewechselt, wenn ihr den Stern.

Game Design

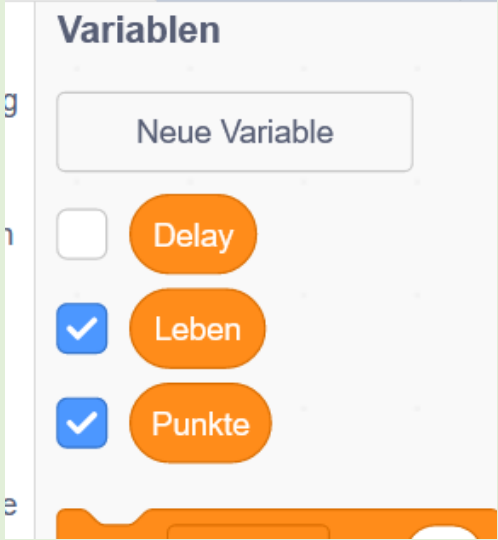


The code is a Scratch script starting with a 'When green flag clicked' event block. It is followed by a 'Repeat' loop block set to 'Repeat forever'. Inside the loop, there are two conditional 'if' blocks. The first 'if' block checks 'if stage image number = 1', and if true, it moves the character to x: -153, y: 76. The second 'if' block checks 'if stage image number = 2', and if true, it moves the character to x: 172, y: -40. The loop ends with a 'Repeat' block arrow.

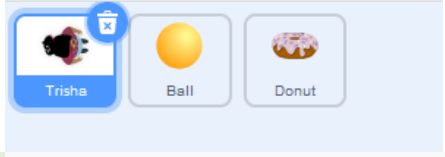
Hier ist der Level-Wechsel Code für den Stern.

Top-Down Shooter

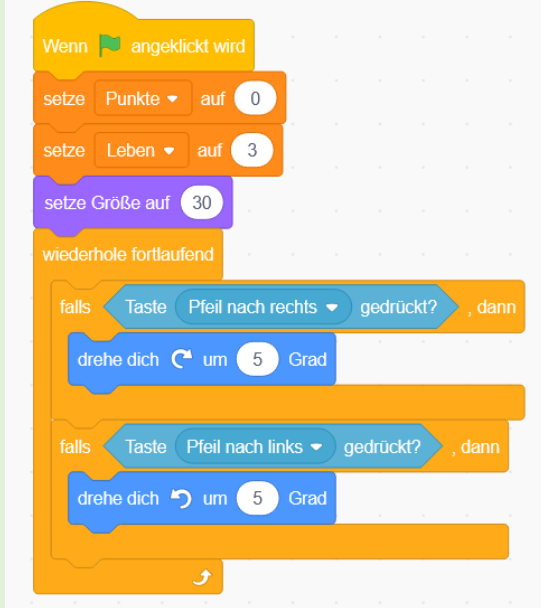
In diesem Spiel müsst ihr die Welt gegen fiese Donuts verteidigen. Ihr steuert euer Raumschiff und schießt auf euch zufliegendes Gebäck ab.



Als erstes müsst ihr drei Variablen erstellen. **Leben** und **Punkte** sollen sichtbar sein.

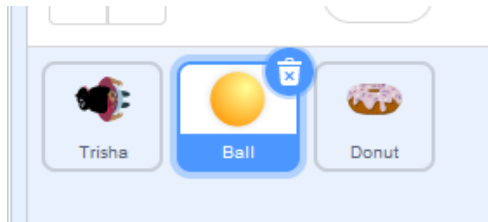


Danach erstellt ihr den Spieler, das Geschoss und die Gegner.



Die Spieler-Figur bekommt den folgenden Code.

Game Design



Wählt jetzt das Geschoss aus.



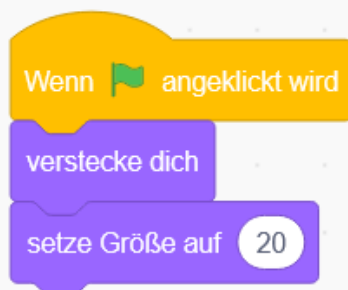
Das Geschoss bekommt diesen code



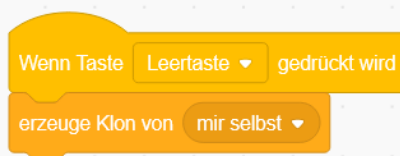
Dieser Code gehört auch zu dem Geschoss. Die Richtung bekommt ihr, wenn ihr



auswählt und die beiden Parameter anpasst.

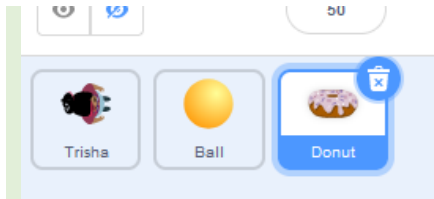


Jetzt noch das Startverhalten des Geschosses.



Und das eigentliche Abfeuern.

Game Design



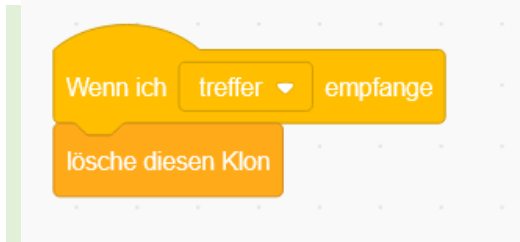
Trisha Ball Donut

Jetzt kommen die Gegner.



```
Wenn Flagge angeklickt wird
  verstecke dich
  setze Größe auf 50
  setze Delay auf 4
  wiederhole fortlaufend
    erzeuge Klon von mir selbst
    warte Zufallszahl von 0.5 bis Delay Sekunden
    falls Delay > 0.5, dann
      ändere Delay um -0.1
```

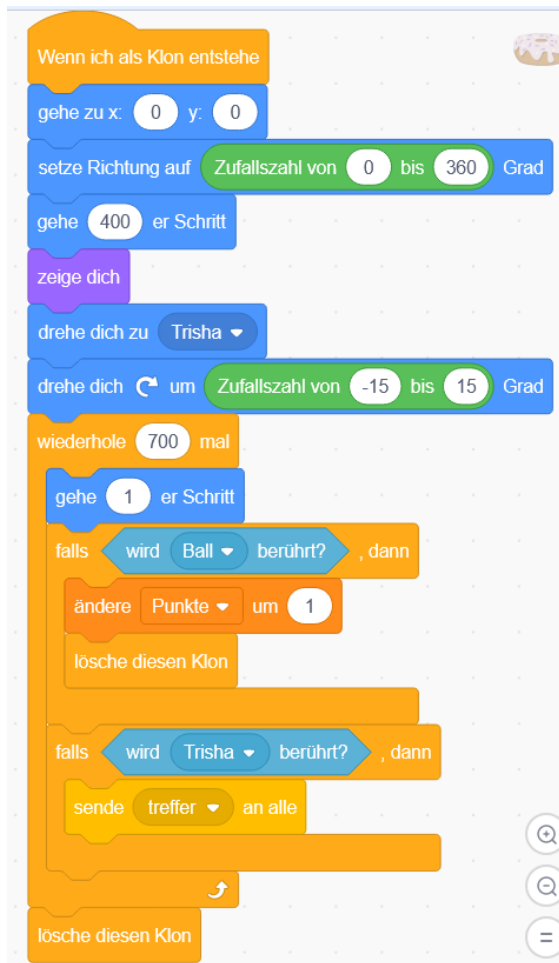
Die Gegner bekommen folgende Code Blöcke.



```
Wenn ich treffer empfangen
  lösche diesen Klon
```

Auch diesen Code.

Game Design



Hier wird die Kopie von einem Gegner bewegt, bis er abgeschossen wird oder den Spieler berührt. Ihr könnt auch die Kostüme wechseln und so die Gegner verändern.

• Interaktive Geschichte

- <https://csfirst.withgoogle.com/c/cs-first/de/geschichten-erz%C3%A4hlen/interaktive-geschichte/einf%C3%BChrung-in-das-interaktive-geschichten-erz%C3%A4hlen.html>

Game Design

Adventure

<https://csfirst.withgoogle.com/c/cs-first/de/game-design/suchspiel/ein-suchspiel-programmieren.html>

Game Design

Und weiter?

Mehr Ideen und Tricks für Spiele findet ihr unter:

[CS First \(https://csfirst.withgoogle.com/c/cs-first/de/game-design/overview.html\)](https://csfirst.withgoogle.com/c/cs-first/de/game-design/overview.html)

Es gibt von CS First aber auch noch viele viele andere Ideen. Schaut mal nach:

<https://csfirst.withgoogle.com/c/cs-first/de/curriculum.html>