



## Scratch - Beispiele für den Unterricht:

Auf unserem Scratch – Padlet findest du

- Unser Scratch - Curriculum
- Arbeitsblätter zum Einsatz im Unterricht
- Links zu interessanten Websites und Padlets



<https://padlet.com/hansvogelmeier/p450jmsn15lt6dlz>

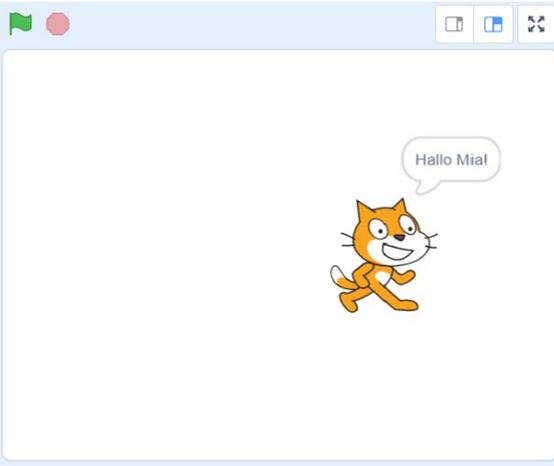
## Lego EV3 - Beispiele für den Unterricht:

Auf unserem Robotic-Padlet findest du

- Anleitung zum Verbinden der EV3-Bausteine mit dem Computer
- Programmbeispiele
- Lego-Mindstorm-Baupläne
- Links zu interessanten Websites



<https://typer.lernwelt-englisch.de/robotik/>



## Du brauchst diese Komponenten:

1 Katze ("Cat1", "Figur1" oder andere Namen)



## Tipp

Im Reiter "**Skripte**" findest du Programmierblöcke mit verschiedenen Anweisungen, die du kombinieren kannst.



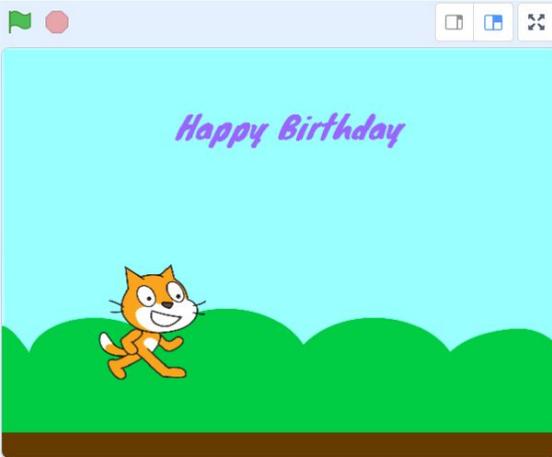
## Aufgabe

1. Gib [scratch.mit.edu](https://scratch.mit.edu) in deinen Browser ein und klicke auf "Entwickeln"  Entwickeln  
→ Dort kannst du programmieren.  
*Hinweis: Um deine Projekte abzuspeichern, musst du dich mit Benutzernamen und Passwort anmelden. Wie das funktioniert, erfährst du von deiner Lehrkraft.*
2. Ändere die **Sprache** bei Bedarf links oben in der Ecke auf **Deutsch**. 
3. Benutze die **Programmierblöcke** unter "Skripte", um die Katze zu animieren:
  - a. **Wenn** man die Katze **anklickt**, soll sie einen **20er-Schritt gehen**.
  - b. Dann soll die Katze "**Hallo [Name]!**" und "**Alles Gute zum Geburtstag!**" **sagen**.
  - c. Lasse die Katze **verschwinden** und **nach 2 Sekunden** wieder **auftauchen**.
4. **Teste** das Ergebnis, indem du auf die Katze klickst.



## Nächste Aufgabe

- Füge ein Hintergrundbild hinzu, indem du ein "Bühnenbild" aus der Bibliothek auswählst.
- Füge auch einen Schriftzug "Happy Birthday" hinzu, indem du selber eine neue Figur zeichnest.



### Du brauchst diese Komponenten:

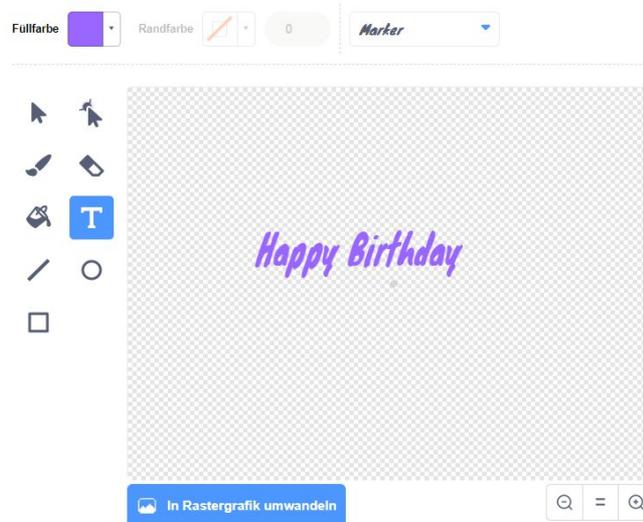
- 1 Bühnenbild ("Blue Sky")
- 1 Figur Schrift (z. B. "Happy Birthday")

### Tipps

Du kannst jede Figur, z. B. die Katze und den Schriftzug, auf der Bühne anklicken und mit gedrückter Maustaste an die gewünschte Position verschieben.

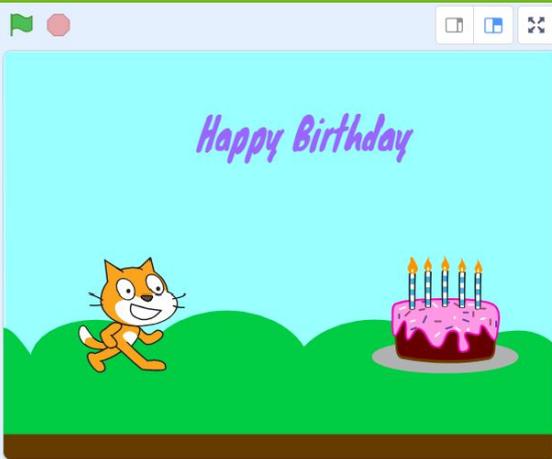
### Aufgabe

1. Füge ein **Hintergrundbild** hinzu, indem du ein "Bühnenbild" aus der Bibliothek auswählst: Klicke dafür unten rechts auf den Button "Bühnenbild wählen"  und suche ein Bild aus der Bibliothek aus.
2. Füge einen **Schriftzug** "Happy Birthday" hinzu, indem du selber eine **neue Figur** zeichnest:
  - a. Positioniere deine Maus dafür auf dem "Figur wählen" Button und klicke anschließend auf "**Malen**". *Hinweis: Mit dem Button "Figur wählen" kannst du neue Figuren hinzufügen. Du kannst sie entweder aus der Figurenbibliothek auswählen, als Bild hochladen oder selber zeichnen.* 
  - b. Benutze das **Textwerkzeug** und schreibe "Happy Birthday".



### Nächste Aufgabe

Füge einen **Geburtstagskuchen** als neue Figur hinzu. Bei "Start" sollen die Kerzen noch aus sein. Wenn er angeklickt wird, sollen die Kerzen brennen und es soll ein "Happy Birthday" Lied abgespielt werden.



### Du brauchst diese Komponenten:

1 Figur Kuchen ("Cake")



### Tipps

Mit der grüne Flagge und dem roten Stoppschild kannst Du deine Animationen und Spiele **starten** und **stoppen**.

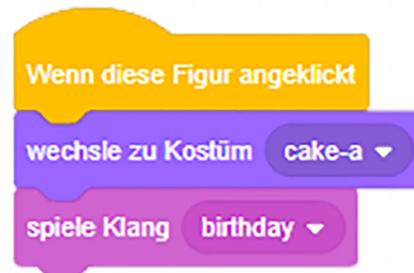


Im Reiter "Klänge" kannst du Klänge aus der **Klangbibliothek** auswählen, eigene aufnehmen, hochladen und bearbeiten.



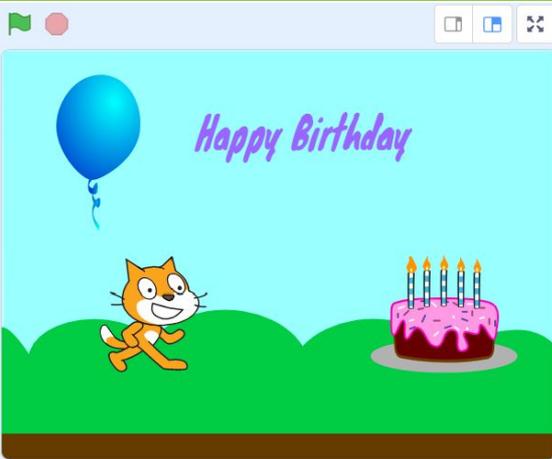
### Aufgabe

1. Füge einen **Geburtstagskuchen** aus der Figurenbibliothek hinzu.
2. Schau dir an, welche **Kostüme** der Kuchen hat. Klicke dafür auf den Kuchen und anschließend auf den Reiter "Kostüme".   
*Hinweis: Du kannst auch selber Kostüme erstellen, hinzufügen und bearbeiten.* 
3. Gehe zurück auf den Reiter "Skripte". **Programmiere den Kuchen:**
  - a. Bei "**Start**" sollen die **Kerzen** noch **aus** sein.
  - b. **Wenn** der Geburtstagskuchen **angeklickt** wird, sollen die **Kerzen brennen** und es soll ein "**Happy Birthday**" **Lied abgespielt werden**.
4. Teste deine fertige Geburtstagskarte, indem Du auf die grüne Fahne und danach auf die verschiedenen Figuren klickst.



### Nächste Aufgabe

Füge auch einen **Luftballon** hinzu. Wenn der Luftballon angeklickt wird, soll sich die Farbe des Ballons ändern.



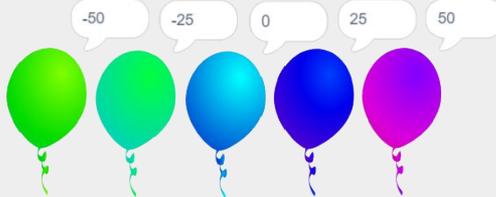
Du brauchst diese Komponenten:

1 Figur Ballon ("Balloon1")



## Tipp

Durch ändern des Farbeffektes um eine bestimmte Zahl, kannst Du die Farbe einer Figur ändern.



## Aufgabe

1. Füge einen **Luftballon** aus der Figurenbibliothek hinzu.
2. **Animiere** deinen Luftballon: **Wenn** der Luftballon **angeklickt** wird, soll sich die Farbe des Ballons ändern. Zum Ändern der Farbe benötigst du einen **"ändere Effekt [Farbe] um"-Block** aus dem Bereich **Aussehen**.
3. **Teste** deine fertige Geburtstagskarte, indem Du auf die grüne Fahne und danach auf die verschiedenen Figuren klickst.



## Nächste Aufgabe

Füge weitere Figuren nach Belieben hinzu und animiere sie (Aussehen, Klang, Bewegung).

## Anmerkung

Wenn der nächste Geburtstag ansteht, kannst Du deine animierte Geburtstagskarte an das Geburtstagskind schicken.

**Hinweis:** Dafür musst Du dich mit *Benutzername* und *Passwort* anmelden. Wie das funktioniert, erfährst Du von deiner Lehrkraft.

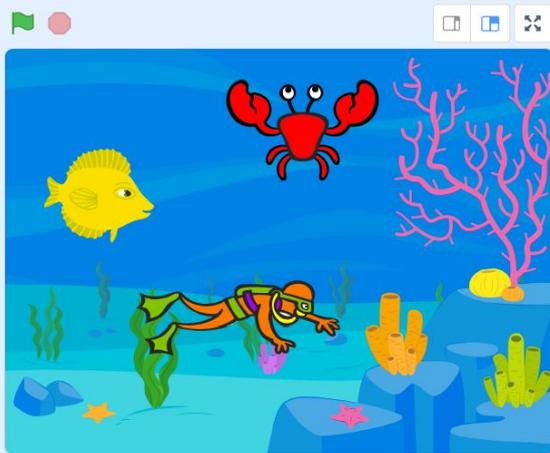
1. Klicke oben rechts auf "Veröffentlichen". Nun siehst Du die fertige Animation.

**Veröffentlichen**

2. Klicke auf das Vollbild-Symbol:



3. Kopiere den Link in der Adresszeile und schicke den Link an das "Geburtstagskind".



### Du brauchst diese Komponenten:

- 1 Bühnenbild Unterwasserwelt ("Underwater 1")
- 1 Figur Krabbe ("Crab")
- 1 Figur Fisch ("Fish")
- 1 Figur Taucher ("Diver2")

### Tipps

Du kannst Figuren per Rechtsklick **löschen** oder, indem du auf das kleine "x" klickst.



Du kannst nach Bühnenbildern sowie Figuren **suchen**.

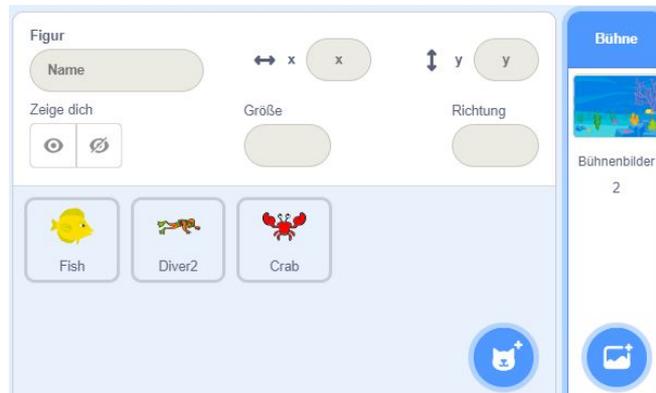


Manchmal gibt es auch das passende Thema, z. B. **"Unterwasserwelt"** bei den Bühnenbildern.



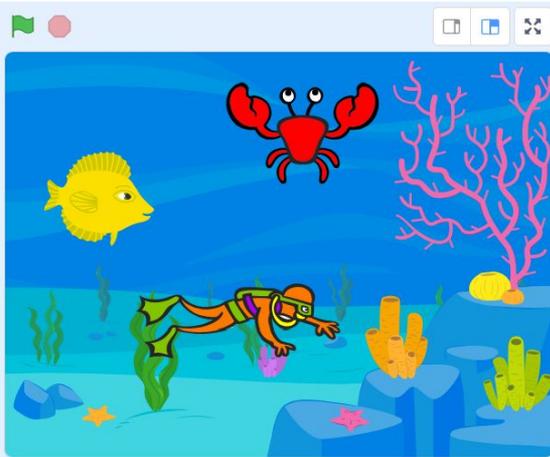
### Aufgabe

- Gib [scratch.mit.edu](https://scratch.mit.edu) in deinen Browser ein und klicke auf "Entwickeln" Entwickeln  
→ Dort kannst du programmieren.  
*Hinweis: Um deine Projekte abzuspeichern, musst du dich mit Benutzernamen 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 das **Bühnenbild** (Unterwasserwelt) aus der Bibliothek aus.
  - Füge alle **Figuren** hinzu: Krabbe, Taucher und Fisch.
  - Ändere, wenn du möchtest, das **Kostüm** für den Fisch zu "fish-d".
  - Lösche die Katze** aus deinem Spiel.



### Nächste Aufgabe

Lasse den **Taucher** in 20er Schritten und den **Fisch** in 30er Schritten immer hin und her schwimmen.



Du brauchst diese Komponenten:



### Tipps

Wenn du Programmierblöcke mit gedrückter Maustaste auf eine andere Figur unten in der Figurenübersicht ziehst, kannst du die **Programmierblöcke** zu der anderen Figur **kopieren**.

Unter "Steuerung" findest Du Programmierblöcke, mit denen Du bestimmte Anweisungen fortlaufend wiederholen kannst.

### Aufgabe

1. Lass den **Taucher** immer hin und her schwimmen:
  - a. Damit der Taucher automatisch entweder nach links oder rechts guckt, je nachdem in welche Richtung er schwimmt, setze **bei Spielstart** den **"Drehtyp" auf links-rechts**.
  - b. **Wiederhole fortlaufend**, dass der Taucher einen **20er Schritt schwimmt** und danach **eine Sekunde Pause** macht.
  - c. Lass den Taucher vom **Rand abprallen**, damit er sofort seine Richtung wechselt und weiter schwimmt, wenn er den Rand berührt.
2. **Erklär** deinem Sitznachbarn/ deiner Sitznachbarin das Programm.
3. Lasse den **Fisch in 30er Schritten** hin und her schwimmen.



### Nächste Aufgabe

Lass die Krabbe bei Spielstart "Fang mich!" sagen. Dann soll die Krabbe immer abwechselnd verschwinden und an einer zufälligen Position wieder erscheinen.



Du brauchst diese Komponenten:



### Tipps

Der Nullpunkt des Koordinatensystems ist genau in der Mitte des Bildes.

Die X-Achse geht von links nach rechts (-240 bis 240) und die Y-Achse geht von oben nach unten (-180 bis 180).

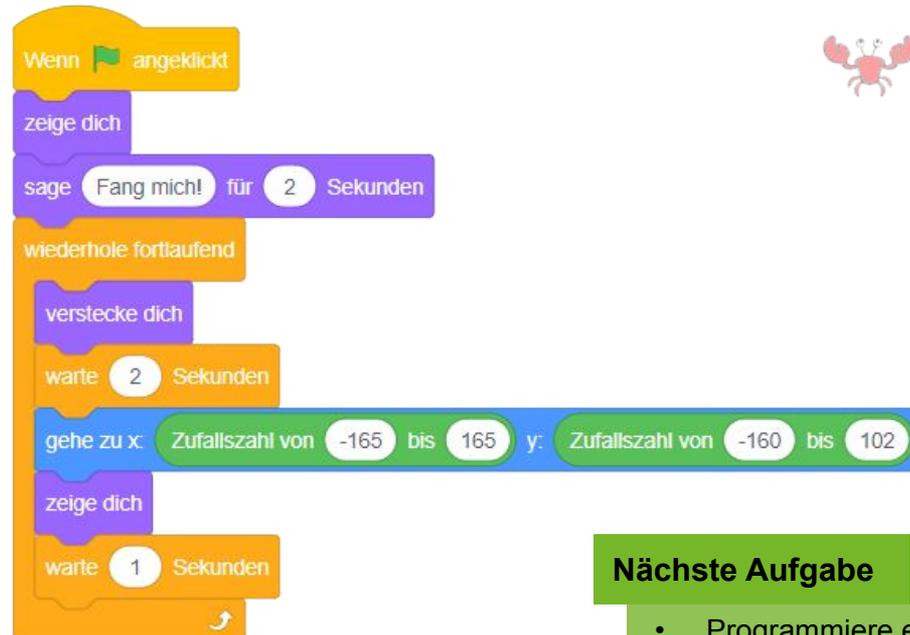
Ziehe die Krabbe mit gedrückter Maustaste in die Ecken der Bühne, um zu prüfen in welchem Bereich sie sichtbar ist. Unter der Bühne werden dir die X- und Y-Position angezeigt.

↔ x 39 ↑ y 161

### Aufgabe

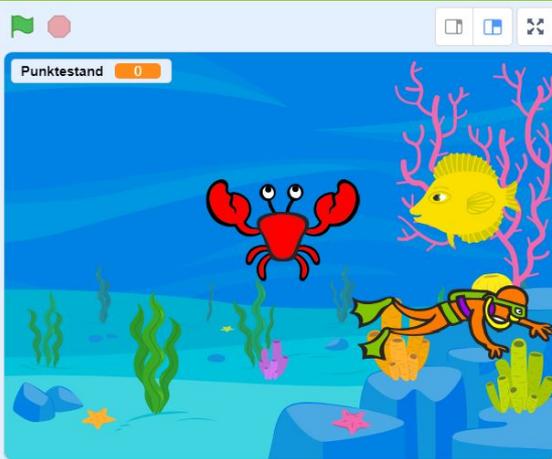
1. Lass die Krabbe **bei Spielstart erscheinen** und **“Fang mich!” sagen**.
2. Lass die Krabbe danach abwechselnd **verschwinden** und an einer **zufälligen Position wieder erscheinen**.

*Hinweis: Um eine zufällige Position zu definieren, benutze den Block "gehe zu..." und füge bei dem X und Y-Wert den Block "Zufallszahl" aus dem Bereich Operatoren ein.*



### Nächste Aufgabe

- Programmiere einen "Punktestand" (Variable) und setze diesen bei Spielstart auf 0.
- Jedes Mal, wenn die Krabbe angeklickt wird, soll ein Klang abgespielt und der Punktestand um 1 erhöht werden.



Du brauchst diese Komponenten:



### Tipp

**Variablen** sind Platzhalter. Sie können beliebige Werte speichern. So z. B. den Punktestand in einem Spiel, der sich häufig verändert.

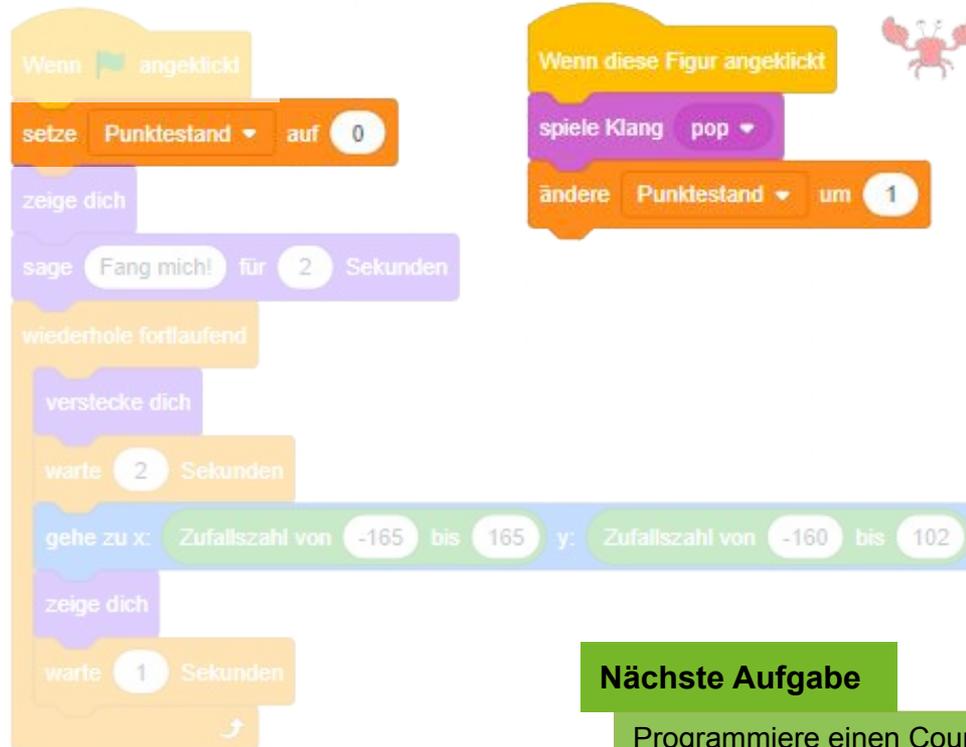


Du kannst festlegen, ob deine Variable auf der Bühne gezeigt wird.



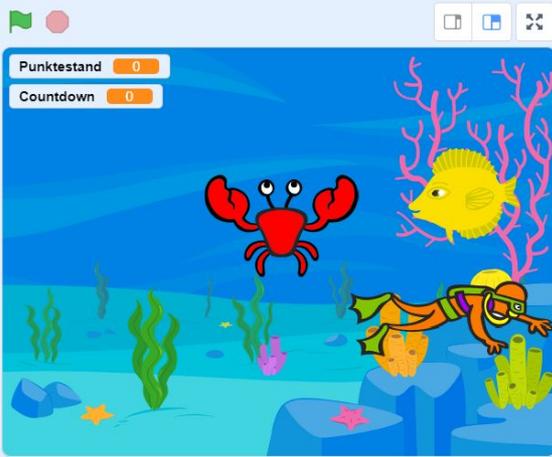
### Aufgabe

1. Programmiere eine **Variable "Punktestand"** und **setze** diesen **bei Spielstart auf 0**. *Hinweis: Links oben im Bildbereich wird nun der Punktestand angezeigt.*
2. Jedes Mal, **wenn die Krabbe angeklickt** wird, lass einen **Klang abspielen** und **erhöhe den Punktestand um 1**.



### Nächste Aufgabe

Programmiere einen Countdown. Jedes Spiel soll nur 20 Sekunden dauern. Wenn die Zeit abgelaufen ist, soll das Spiel gestoppt werden. **Tipp:** Für den Countdown benötigst du noch eine neue Variable.



Du brauchst diese Komponenten:



### Tipp

Mit der grünen Flagge und dem roten Stoppschild kannst Du deine Animationen und Spiele **starten** und **stoppen**.



### Info

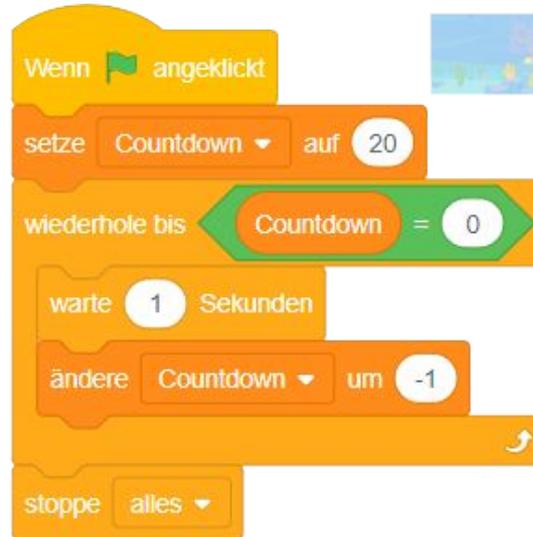
Mit dieser Befehlsfolge hast Du deinen eigenen **Countdown** gebaut. Links oben im Bildbereich wird nun immer die verbleibende Zeit angezeigt.

**Achtung!** Die Zeit läuft - jedes Spiel dauert nur noch 20 Sekunden.

### Aufgabe

1. **Klicke** auf die **Bühne** und öffne den Reiter Skripte.   
2. **Programmiere einen Countdown:**
  - a. Erstelle eine neue **Variable "Countdown"**.
  - b. **Setze** den Countdown bei Spielstart **auf 20**.
  - c. **Ändere die Variable jede Sekunde um -1**. Wenn die **Zeit abgelaufen** ist (= 0), **stoppe das gesamte Spiel**.

*Hinweis: Unter "Operatoren" findest du Code-Blöcke für mathematische Gleichungen wie z.B. =, <, >.*
3. **Erklär** deinem Sitznachbarn/ deiner Sitznachbarin das Programm. Wechselt euch mit dem Erklären ab.



### Nächste Aufgabe

Füge einen Hai hinzu, der wie die Krabbe an zufälligen Positionen erscheint. Klickt man versehentlich den Hai an, soll sich der Punktestand um -1 verringern.



### Du brauchst diese Komponenten:

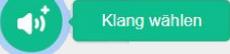
1 Figur Hai ("Shark 2")



### Tipps

Wenn Du **Programmierblöcke** mit gedrückter Maustaste auf eine andere Figur unten in der Figurenübersicht ziehst, kannst Du sie dorthin **kopieren**.

Im Reiter **Klänge** kannst du einen Klang aus der Klangbibliothek auswählen. Du kannst auch Eigene aufnehmen, hochladen und die Klänge bearbeiten.



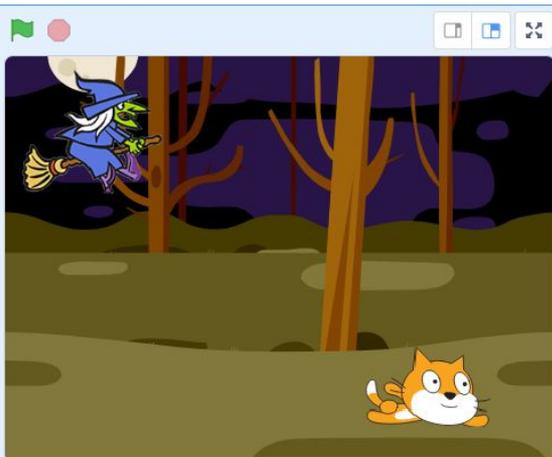
### Aufgabe

1. Füge einen **Hai** aus der Bibliothek hinzu. Der Hai soll sich so verhalten wie die Krabbe: Er soll sich **verstecken** und an **anderer Stelle** wieder **auftauchen**.
2. **Wenn der Hai angeklickt** wird, soll folgendes passieren:
  - a. **Ändere den Punktestand um -1**.
  - b. Öffne den Reiter "**Klänge**" und suche den Klang "**Drum Boing**".
  - c. Gehe zurück in den Reiter Skripte und spiele statt "pop" den **Schlagzeugsound**.
3. **Teste dein Spiel**, indem du auf die grüne Fahne klickst.



### Nächste Aufgabe

- Überlege dir, wie Du das Spiel noch anpassen kannst.
- Verhindere z. B., dass sich nach Ablauf der Zeit die Punkte durch Klick auf die Krabbe bzw. den Hai verändern können.



### Du brauchst diese Komponenten:

- 1 Bühnenbild Zauberwald (“Woods”)
- 1 Figur Katze (“Cat Flying”)
- 1 Figur Hexe ([bit.ly/figur-hexe](http://bit.ly/figur-hexe))



### Tipp

Du kannst eigene Figuren in deinem Projekt **hochladen**.

Unterhalb der Bühne kannst du Eigenschaften der Figuren ändern. Zum Beispiel die **Größe der Hexe**.



### Aufgabe

- Gib [scratch.mit.edu](http://scratch.mit.edu) in deinen Browser ein und klicke auf “Entwickeln”  Entwickeln  
→ Dort kannst du programmieren.  
*Hinweis: Um deine Projekte abzuspeichern, musst du dich mit Benutzernamen 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 das **Bühnenbild** (“Woods”) als Hintergrund aus der Bibliothek aus.
  - Lösche die Katze** und füge stattdessen die **fliegende Katze als Figur** hinzu.
  - Lade die Hexe als neue Figur** in dein Projekt. Mache die Hexe etwas kleiner.
- Bei Spielstart** soll die Hexe **oben links** in der Ecke **erscheinen**. Lass die Hexe **hin und her fliegen**. *Hinweis: Damit die Hexe schräg nach unten und oben fliegen kann, drehe die Hexe, z. B. um 15 Grad.*



### Nächste Aufgabe

Bei Spielstart soll die Katze unten rechts im Bild erscheinen. Bewege die Katze im Spiel mit den Pfeiltasten.



## Aufgabe

1. Bei Spielstart soll die fliegende Katze unten rechts erscheinen. Setze das **Kostüm** auf "cat flying-b".
2. Bewege die Katze mit den Pfeiltasten.
  - a. Prüfe mit Hilfe von bedingten Anweisungen (**falls... dann...**), ob eine bestimmte **Taste gedrückt** wird.
  - b. **Drehe die Katze**, sodass die Katze nach links guckt, wenn sie sich nach links bewegt und nach rechts guckt, wenn sie sich nach rechts bewegt.
  - c. Lass die Katze **vom Rand "abprallen"** und setze den **Drehtyp auf links-rechts**, damit sie sofort ihre Richtung wechselt, wenn sie den Rand berührt.

## Du brauchst diese Komponenten:



## Tipps

Bisher haben wir Figuren mit dem Block "gehe ...-Schritte" bewegt. Eine andere Möglichkeit **Figuren zu bewegen** ist, deren X und Y-Werte im Koordinatensystem zu ändern.

- nach rechts: positiver X-Betrag
- nach links: negativer X-Betrag
- nach oben: positiver Y-Betrag
- nach unten: negativer Y-Betrag

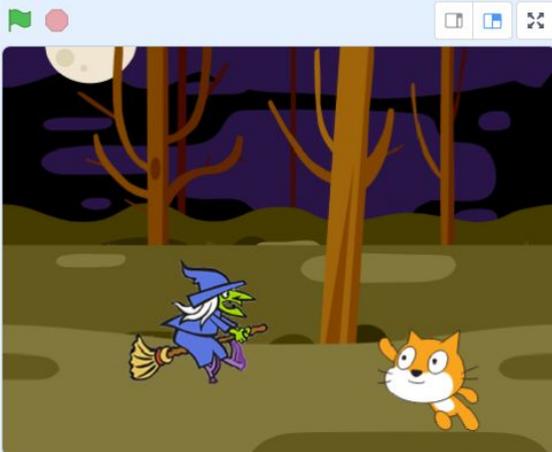
Wenn du eine falls-dann Anweisung fertig hast, kannst du diese einfach mit Rechtsklick **duplizieren** und entsprechend anpassen.

The code blocks are as follows:

- Wenn **angeklickt**
- wechsele zu Kostüm **cat flying-b**
- gehe zu x: **129** y: **-113**
- setze Drehtyp auf **links-rechts**
- wiederhole fortlaufend
  - falls **Taste Pfeil nach oben gedrückt?** , dann
    - ändere y um **10**
  - falls **Taste Pfeil nach unten gedrückt?** , dann
    - ändere y um **-10**
  - falls **Taste Pfeil nach rechts gedrückt?** , dann
    - ändere x um **10**
    - setze Richtung auf **90** Grad
  - falls **Taste Pfeil nach links gedrückt?** , dann
    - ändere x um **-10**
    - setze Richtung auf **-90** Grad
  - pralle vom Rand ab

## Nächste Aufgabe

Wenn die Hexe die Katze berührt, soll sie den Zauberspruch "Abrakadabra" sagen und einen Klang abspielen.



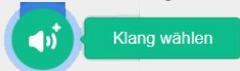
Du brauchst diese Komponenten:



## Tipps

Überprüfe mit Hilfe einer bedingten Anweisung (falls... dann...), ob die Hexe die Katze berührt.

Wähle im Reiter **Klänge** einen Klang aus der Klangbibliothek aus.



Dann kannst du beim Block "spiele Klang" den Klang auswählen.



## Aufgabe

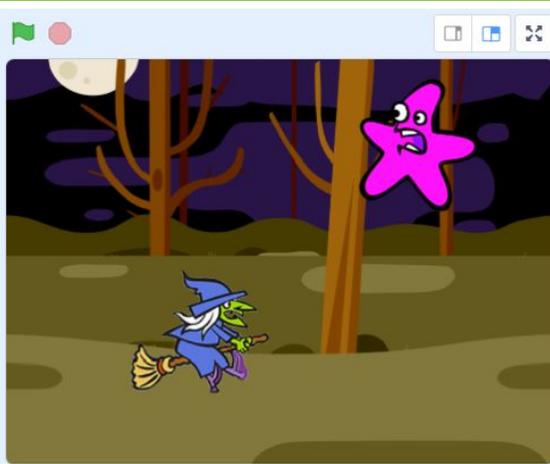
- Jedes Mal**, wenn die **Hexe die Katze berührt**, soll folgendes passieren:
  - Die Hexe soll einen Zauberspruch, z. B. **"Abrakadabra"** sagen.
  - Es soll ein **Klang abgespielt** werden, z. B. "Fairydust".
- Erkläre** deiner Programmierpartnerin / deinem Programmierpartner den Code.



## Nächste Aufgabe

Sobald die Hexe die Katze berührt, soll sie eine Nachricht "verzaubert" senden. Wenn die Katze diese Nachricht empfängt, soll sie das Kostüm wechseln und sich in ein anderes zufälliges Tier verwandeln.

**Tip:** Füge dazu neue Kostüme aus der Bibliothek hinzu und erstelle eine neue Variable "Zufallskostüm".



Du brauchst diese Komponenten:



### Tip

Füge für die Katze im Reiter Kostüme

neue Kostüme/ Tiere aus der Bibliothek hinzu.



Hinweis: Gegebenenfalls ist es sinnvoll, die Größen anzupassen.

### Aufgabe

Verzaubere die Katze in ein zufälliges Tier. Gehe dazu wie folgt vor:

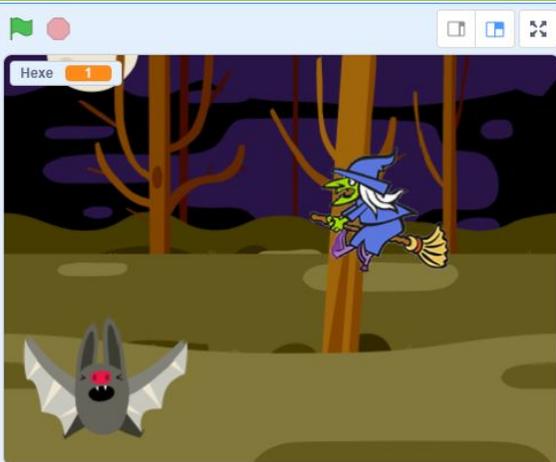
1. Füge bei der Katze im Reiter **Kostüme** vier neue Tiere **aus der Bibliothek** hinzu.
2. Gehe zurück in den Reiter "Skripte". Erstelle eine neue **Variable "Zufallskostüm"**.  
*Hinweis: Auf der Bühne wird automatisch die Variable angezeigt. Durch Abwählen der Checkbox kannst du sie ausblenden:*  Zufallskostüm
3. Wenn die Hexe die Katze berührt, sende eine neue **Nachricht "verzaubert" an alle**.  
*Hinweis: Den "sende Nachricht"-Block findest du unter Ereignisse.*
4. Wenn die Katze die **Nachricht "verzaubert" empfängt**, setze die **Variable** auf eine **Zufallszahl** von 1 bis 6 und **wechsle das aktuelle Kostüm** zum **"Zufallskostüm"**.



### Nächste Aufgabe

Programmiere einen Punktestand. Jedes Mal, wenn die Hexe die Katze verzaubert, bekommt sie einen Punkt.

**Tip:** Erstelle eine neue Variable.



Du brauchst diese Komponenten:



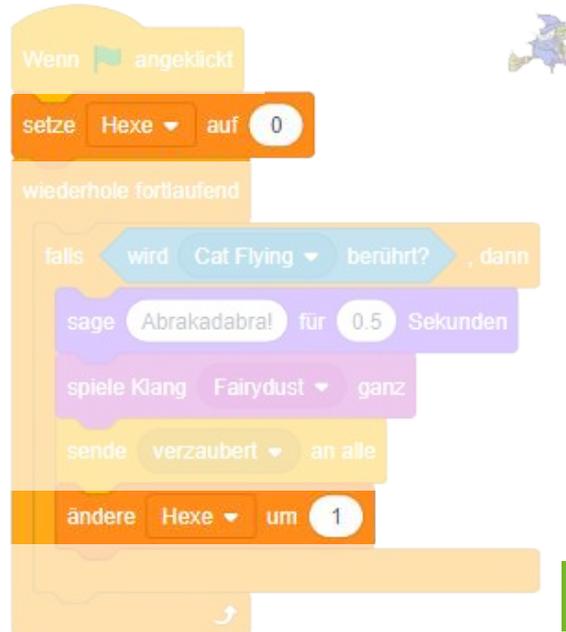
## Tipp

Wenn dir das Spiel zu schwierig ist, kannst du nach dem Ändern der Variable noch einen **warte-Block** hinzufügen.

Wenn dir das Spiel hingegen zu einfach ist, kannst du die Schritte der Hexe vergrößern oder die Figuren größer machen.

## Aufgabe

1. Erstelle eine neue **Variable "Hexe"**. Setze die Variable **bei Spielstart auf 0**.
2. **Ändere** die Variable jedes Mal **um 1**, wenn die Hexe den Spieler berührt.  
*Hinweis: Die Katze ist der Spieler. Wenn sie verzaubert ist, sieht sie allerdings nicht mehr aus wie eine Katze.*



## Nächste Aufgabe

Auch du sollst Punkte bekommen. Jedes Mal einen Punkt, wenn du (der Spieler) 5 Sekunden lang nicht verzaubert wirst.

**Tipp:** Erstelle dafür eine weitere Variable und benutze die "Stoppuhr".



Du brauchst diese Komponenten:



### Hinweis

In diesem Beispiel liegt der Code auf der **Bühne**. Alternativ kannst du auch den Code bei der Katze ergänzen.

### Aufgabe

1. Erstelle eine neue **Variable "Spieler"**. Setze die Variable **bei Spielstart auf 0**.
2. Benutze die **Stoppuhr**, um die Zeit zu messen.
  - a. Immer, **wenn** du (der Spieler) 5 Sekunden lang nicht verzaubert wirst (**Stoppuhr > 5**), **erhöhe** deinen Punktstand (Variable Spieler) **um 1** und **setze die Stoppuhr zurück** auf 0.
  - b. Auch bei Spielstart und wenn die Hexe dich verzaubert, muss jedes Mal die Stoppuhr zurückgesetzt werden, damit die 5 Sekunden wieder neu gezählt werden.

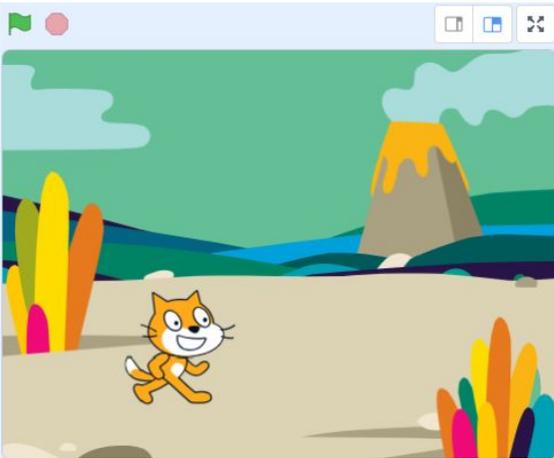
*Hinweis: Im Bereich Fühlen findest du alle Blöcke für die Stoppuhr. Die Stoppuhr startet immer automatisch bei Spielstart und zählt die Sekunden.*

3. **Teste dein Spiel**, indem du auf die grüne Fahne klickst.



### Nächste Aufgabe

- Überlege dir, wie Du das Spiel noch anpassen kannst.
- Definiere z. B., wann du das Spiel verlierst und unter welchen Bedingungen du gewinnst.



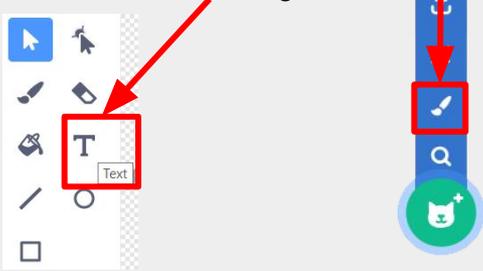
### Du brauchst diese Komponenten:

- 1 Bühnenbild (z. B. "Jurassic")
- 1 Katze ("Cat1", "Figur1" oder andere Namen)



### Tip

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



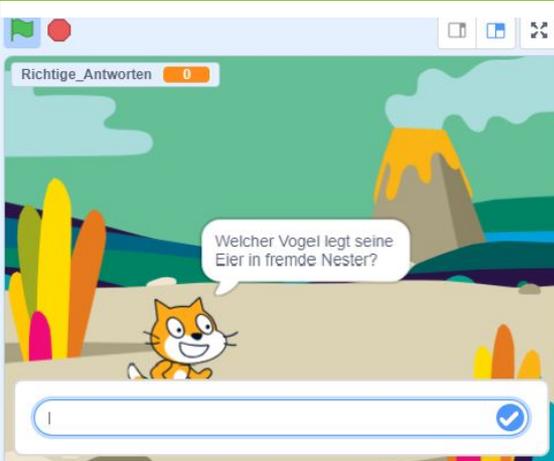
### Aufgabe

1. Gib [scratch.mit.edu](https://scratch.mit.edu) in deinen Browser ein und klicke auf "Entwickeln"  → Dort kannst du programmieren.  
*Hinweis: Um deine Projekte abzuspeichern, musst du dich mit Benutzernamen und Passwort anmelden. Wie das funktioniert, erfährst du von deiner Lehrkraft.*
2. Ändere die **Sprache** bei Bedarf auf **Deutsch**. 
3. Links siehst du das Design des Spiels und die Komponenten, die du benötigst.
  - a. Wähle ein **Bühnenbild** aus der Bibliothek als Hintergrundbild aus.
  - b. Überlege dir ein **Thema für dein Quiz**, z. B. "Tiere". Füge deinen **Quiznamen** als Schriftzug hinzu.
4. 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.  
**Tip:** Eigene Blöcke kannst du im Bereich "Meine Blöcke" erstellen.
- Definiere deinen Block: Setze die Variable auf 0 und stelle eine erste Quizfrage.

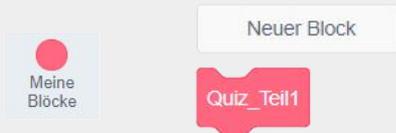


Du brauchst diese Komponenten:



### 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.



Du brauchst diese Komponenten:



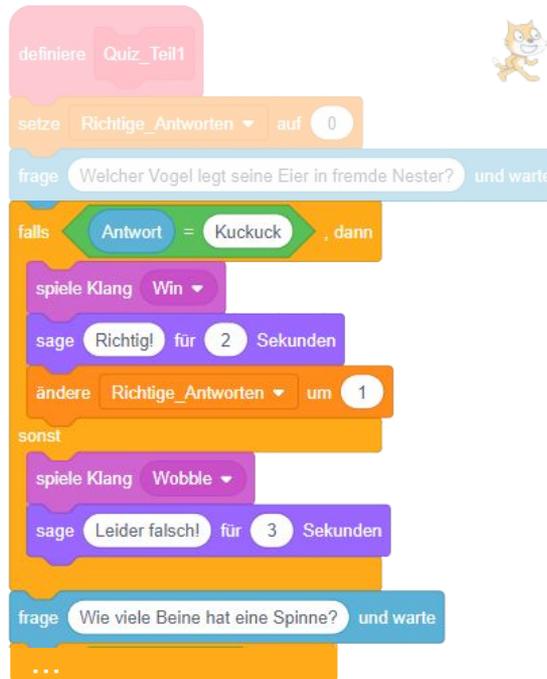
### Tipp

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

- Überprüfe die eingegebene Antwort. Je nachdem, ob die Antwort richtig oder falsch ist, soll die Katze unterschiedlich reagieren:
  - Wenn** die eingegebene **Antwort richtig ist**, spiele einen **Klang** ab, lass die Katze **"Richtig!" sagen** und **ändere** die **Variable "Richtige\_Antworten" um 1**.
  - 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.*
- Füge zwei weitere **Fragen** hinzu und überprüfe jeweils die eingegebenen Antworten.
- 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".



Du brauchst diese Komponenten:

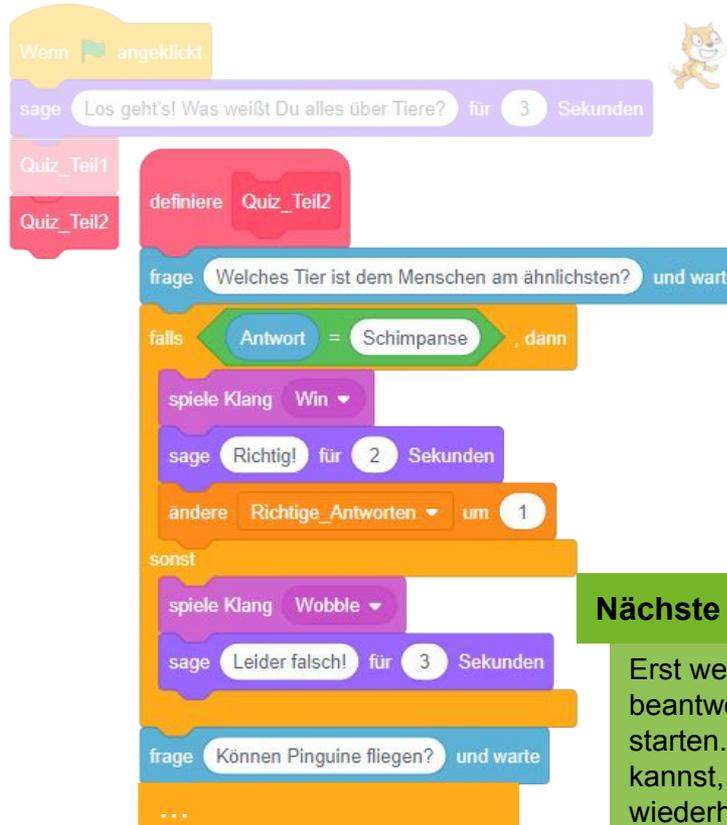


### Hinweis

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.



Du brauchst diese Komponenten:



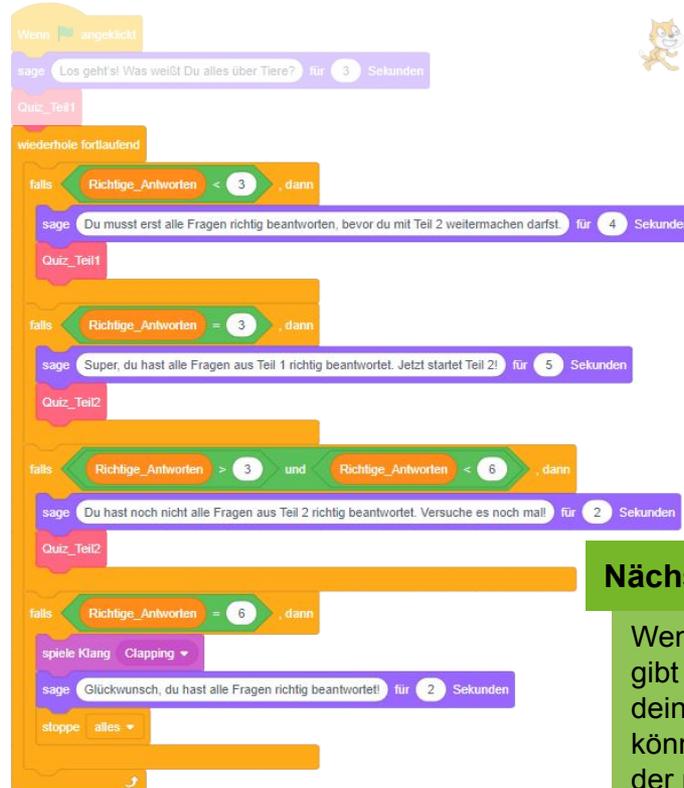
## Tip

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.



### 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:



```
Richtige_Antworten = 6 oder Richtige_Antworten > 6
```

### 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
setze Richtige_Antworten auf 3
frage Welches Tier ist dem Menschen am ähnlichsten? und warte
falls Antwort = Schimpanse dann
  spiele Klang Win
  sage Richtig! für 2 Sekunden
  ändere Richtige_Antworten um 1
...
```

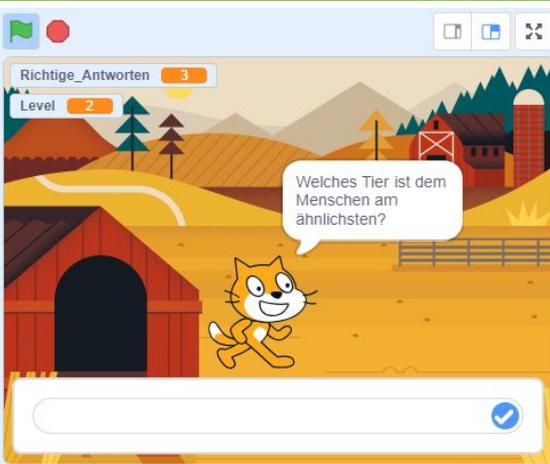


**Erläuterung:** Warum müssen wir die Variable `Richtige_Antworten` in Quiz\_Teil2 auf den Wert = 3 setzen?

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.



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 anzeigt, 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.*

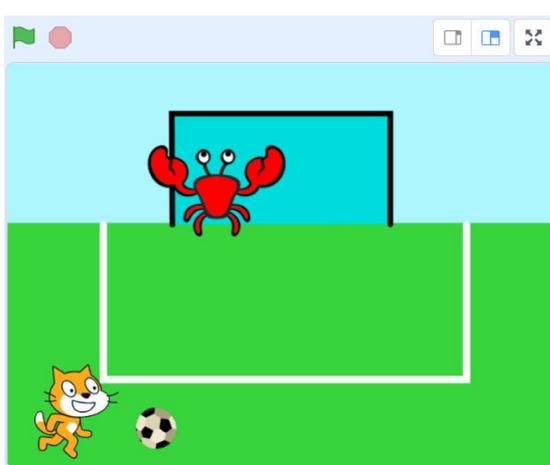
```
definiere Quiz_Teil1
setze Richtige_Antworten auf 0
setze Level auf 1
wechsle zu Bühnenbild Jurassic
```

```
definiere Quiz_Teil2
setze Richtige_Antworten auf 3
setze Level auf 2
wechsle zu Bühnenbild Farm
```



### Nächste Aufgabe

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



### Du brauchst diese Komponenten:

- 1 Figur Katze ("Cat")
- 1 Figur Krabbe ("Crab")
- 1 Figur Fußball ("Soccer Ball")
- 1 Figur Tor (selbst malen)
- 1 Bühnenbild Fußballfeld (malen)

### Tipp

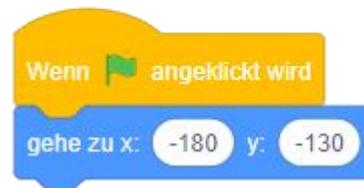
Um einen neuen **Hintergrund** zu **erstellen**, fahre auf den Kreis für "Bühnenbild wählen" und klicke darüber auf den **Pinsel** "Malen".

Du kannst eine eigene **Figur zeichnen**, indem du über den Kreis "Figur wählen" fährst und dann auf den **Pinsel** für "Malen" klickst.



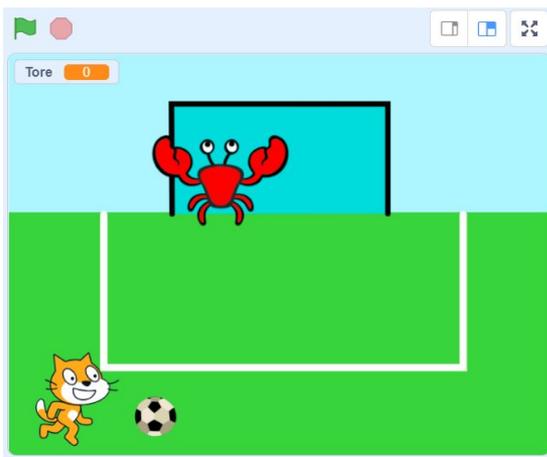
### Aufgabe

1. Gib [scratch.mit.edu](https://scratch.mit.edu) in deinen Browser ein und klicke auf "Entwickeln"  → Dort kannst du programmieren.  
*Hinweis: Um deine Projekte abzuspeichern, musst du dich mit Benutzernamen und Passwort anmelden. Wie das funktioniert, erfährst du von deiner Lehrkraft.*
2. Ändere die **Sprache** bei Bedarf links oben in der Ecke auf **Deutsch**. 
3. Erstelle einen neuen **Hintergrund**: Zeichne eine grüne Fußballwiese mit blauem Himmel, aber noch **kein** Tor.
4. Füge neue Figuren hinzu und positioniere die Figuren auf der Bühne:
  - a. Füge ein **Tor** hinzu, indem du selbst eine neue Figur zeichnest.
  - b. Füge die **Krabbe** und einen **Fußball** aus der Figurenbibliothek hinzu.
  - c. Aktiviere unter "Kostüme" das andere Kostüm der **Katze** ("Kostüm2").
  - d. Mache deine **Spielfiguren kleiner** und positioniere sie wie links in der **Designvorschau**.
5. **Programmiere**, dass die Katze bei jedem **Spielstart** am unteren **Spielrand** erscheint.



### Nächste Aufgabe

- Steuere die Katze mit den Pfeiltasten nach links, rechts, oben und unten.
- Animiere die Torhüter-Krabbe. Sie soll nach Spielstart immer wieder an eine zufällige Position zwischen dem linken und rechten Torpfosten gleiten.
- Definiere eine Variable "Tore" und setze sie bei Spielstart auf 0.



## Aufgabe

1. Steuere die Katze mit den **Pfeiltasten** nach links, rechts, oben und unten, indem du die **X- und Y-Werte** änderst. Sorge außerdem dafür, dass sie in die **Richtung** schaut, in die sie läuft.
2. Animiere die Torhüter-Krabbe. Die Krabbe soll nach **Spielstart fortlaufend** an eine **zufällige** Position zwischen den Torpfosten **gleiten**.
3. Der Fußball soll bei **Spielstart** am unteren **Spielrand** erscheinen. Definiere eine **Variable "Tore"** und setze sie bei **Spielstart** auf 0.

```

Wenn [angeklickt wird]
  gehe zu x: -180 y: -130
  setze Drehtyp auf links-rechts
  setze Richtung auf 90 Grad
  wiederhole fortlaufend
    falls [Taste Pfeil nach links gedrückt?] dann
      setze Richtung auf -90 Grad
      ändere x um -5
    falls [Taste Pfeil nach rechts gedrückt?] dann
      setze Richtung auf 90 Grad
      ändere x um 5
    falls [Taste Pfeil nach oben gedrückt?] dann
      ändere y um 5
    falls [Taste Pfeil nach unten gedrückt?] dann
      ändere y um -5
  pralle vom Rand ab
  
```



```

Wenn [angeklickt wird]
  wiederhole fortlaufend
    gleite in 1 Sek. zu x: Zufallszahl von -80 bis 80 y: 40
  
```



```

Wenn [angeklickt wird]
  setze Tore auf 0
  gehe zu x: -110 y: -145
  
```



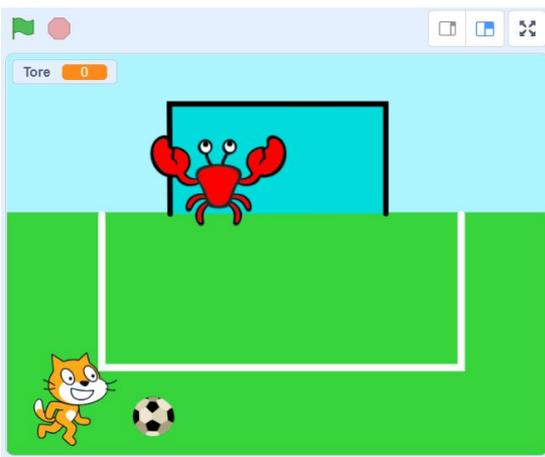
**Beachte:** Je nachdem wie groß dein Tor ist, musst du diese Werte anpassen.

## Tipp

- Beachte Folgendes, wenn du Figuren bewegen möchtest:
- nach rechts: positiver X-Wert
  - nach links: negativer X-Wert
  - nach oben: positiver Y-Wert
  - nach unten: negativer Y-Wert

## Nächste Aufgabe

- Falls die Katze den Fußball berührt, bewege den Ball in Richtung Tor.
- Falls der Fußball die Krabbe berührt, spiele einen Klang und lasse ihn an eine zufällige Stelle am unteren Bildrand gleiten.



### Tipps

Bevor du einen Klang verwenden kannst, musst du ihn hinzufügen. Wähle dazu oben den Reiter **Klänge** aus. Dann kannst du unten links einen Klang aus der Klangbibliothek auswählen.

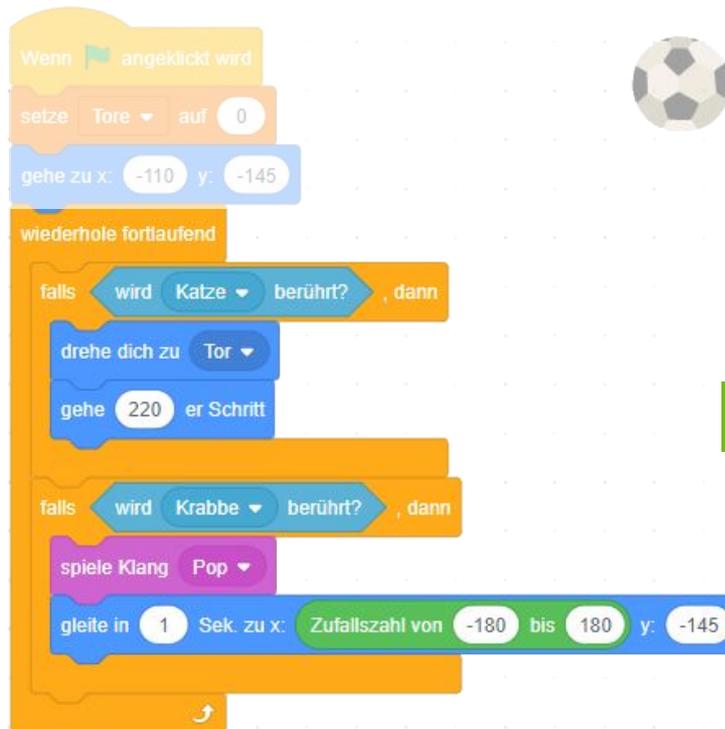


Durch das **Senden und Empfangen von Nachrichten**, können die Aktionen mehrerer Figuren koordiniert werden. Du findest die Anweisung *“sende Nachricht an”* unter Ereignisse.

### Aufgabe

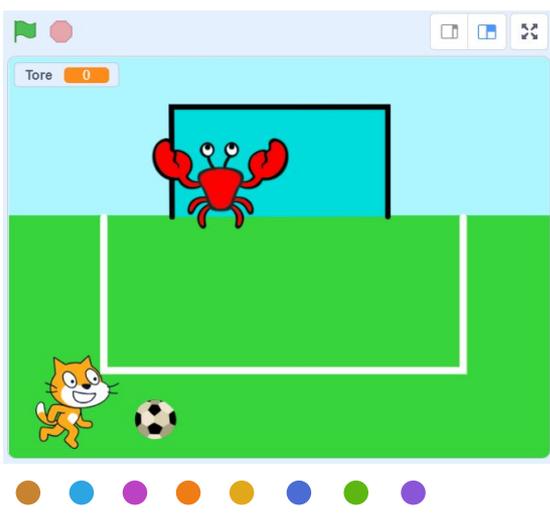
#### Animiere den Fußball:

1. **Falls** die Katze den Fußball **berührt**, **drehe ihn** zum Tor und **gehe** mit dem Fußball 220 Schritte. *Hinweis: Eventuell musst du die Anzahl der Schritte anpassen, damit der Fußball das Tor erreicht.*
2. **Falls** die Krabbe den Fußball **berührt**, spiele einen **Klang** und lasse ihn an eine **zufällige** Stelle am unteren Bildrand **gleiten**.



### Nächste Aufgabe

Falls der Fußball das Tor berührt (Farbe!), dann ändere die Variable "Tore" um 1 und sende eine Nachricht "Tor" an alle. Spiele einen Klang ab und lass den Fußball an eine zufällige Position am unteren Spielrand gleiten.



## Tip

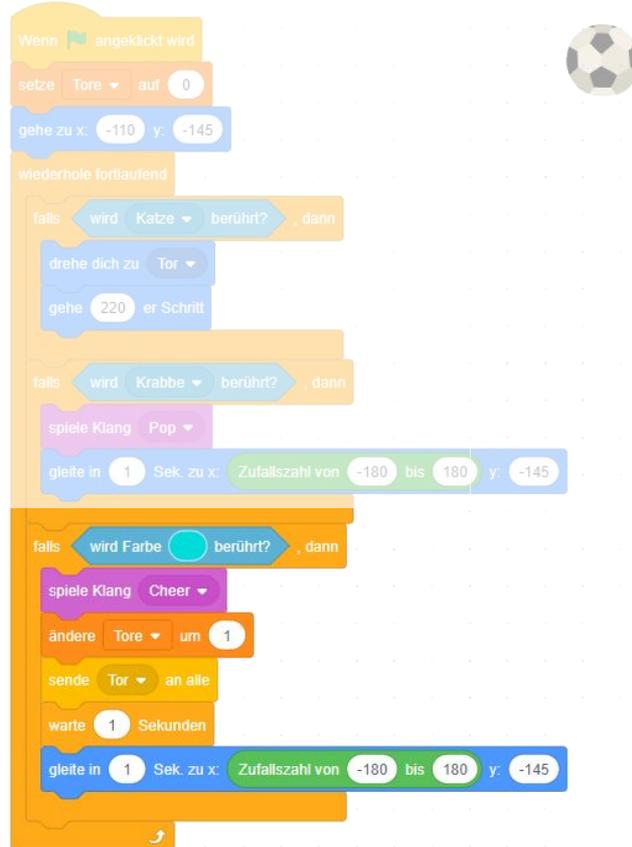
Um eine bestimmte Farbe als Bedingung auszuwählen, klicke zuerst in das Farbfeld, danach auf die Pipette und dann an eine Stelle im Bildbereich, um die Farbe aufzunehmen.



## Aufgabe

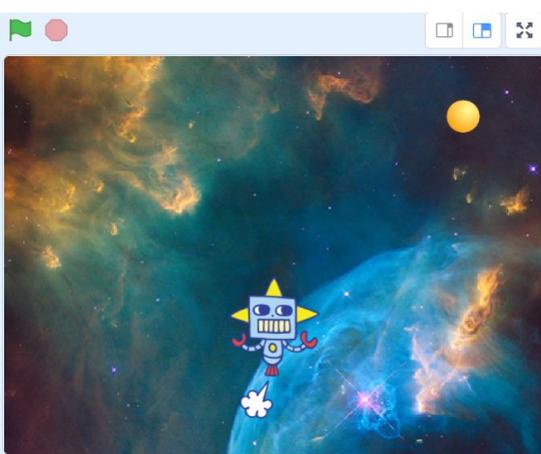
Animiere den Fußball:

- Falls** der Fußball das Tor **berührt** (Farbe), dann **ändere die Variable** "Tore" um 1 und **sende** eine **Nachricht** "Tor" an alle. **Spiele** einen **Klang** ab und lass den Fußball an eine **zufällige** Position am unteren Spielrand **gleiten**.
- Wenn** die Katze die Nachricht "Tor" **empfängt**, soll sie "TOOR!!" **sagen**.



## Nächste Aufgabe

Überlege dir, wie du das Spiel noch weiter optimieren kannst. Zum Beispiel kannst du den Fußball so programmieren, dass er den Himmel berührt und danach an eine zufällige Position am unteren Spielrand zurückrollt.



## Du brauchst diese Komponenten:

- 1 Figur Roboter ("Robot")
- 1 Figur Ball ("Ball")
- 1 Bühnenbild Weltraum ("Nebula")



## Tipp

Mache den Roboter und den Ball kleiner, damit das Spiel schwieriger wird.

Die **Größe** deiner Figuren, kannst du unterhalb der Bühne anpassen.

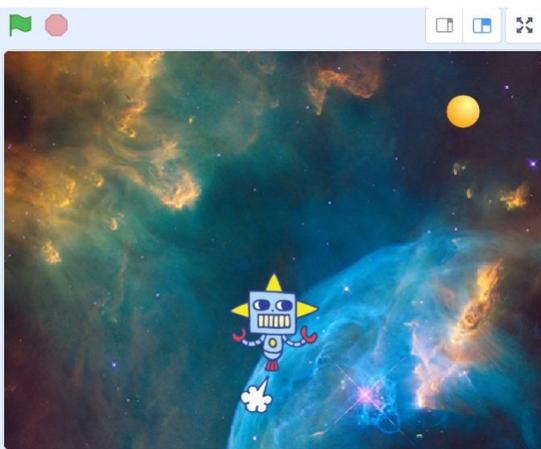
## Aufgabe

1. Gib [scratch.mit.edu](https://scratch.mit.edu) in deinen Browser ein und klicke auf "Entwickeln"  Entwickeln  
→ Dort kannst du programmieren.  
*Hinweis: Um deine Projekte abzuspeichern, musst du dich mit Benutzernamen und Passwort anmelden. Wie das funktioniert, erfährst du von deiner Lehrkraft.*
2. Ändere die **Sprache** bei Bedarf links oben in der Ecke auf **Deutsch**. 
3. Wähle das **Bühnenbild** "Nebula" als Hintergrund aus der Bibliothek aus und füge den Roboter und den Ball als neue **Figuren** hinzu.
4. Entferne die Figur "Katze".
5. **Programmiere**, dass der Roboter bei **Spielstart** am unteren **Spielrand** erscheint.



## Nächste Aufgabe

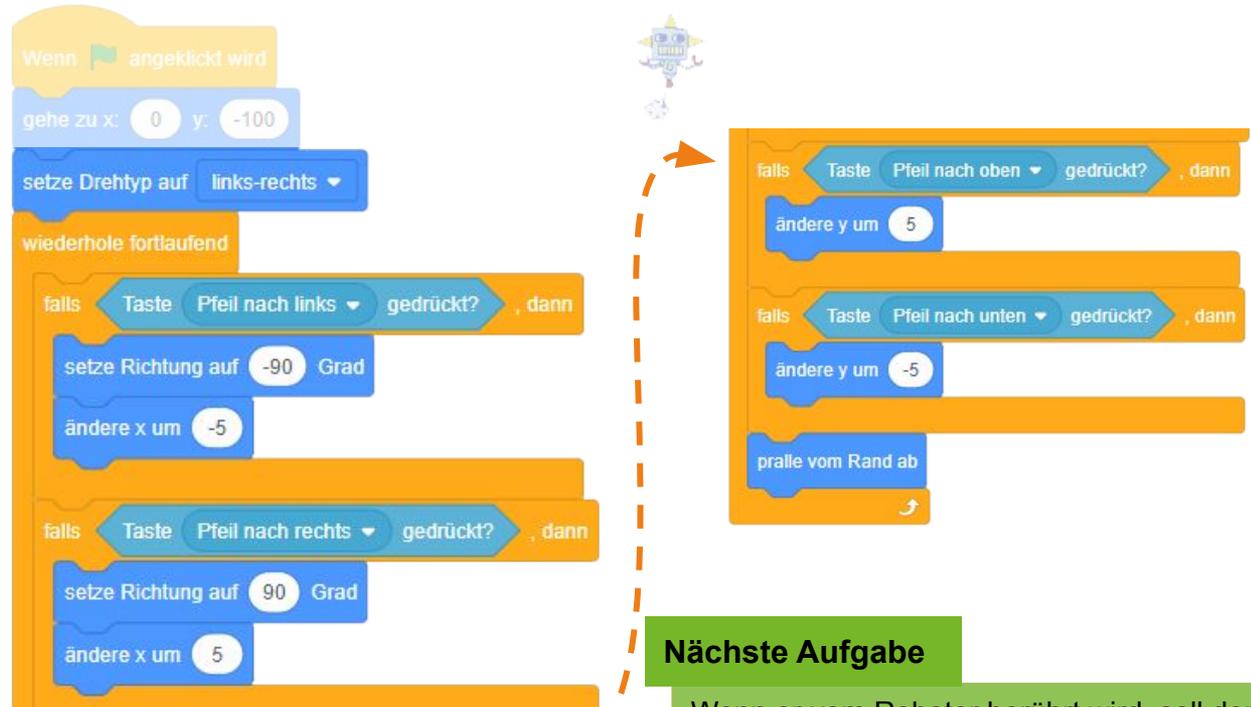
Bewege den Roboter mit den Pfeiltasten nach rechts, links, oben und unten.



### Aufgabe

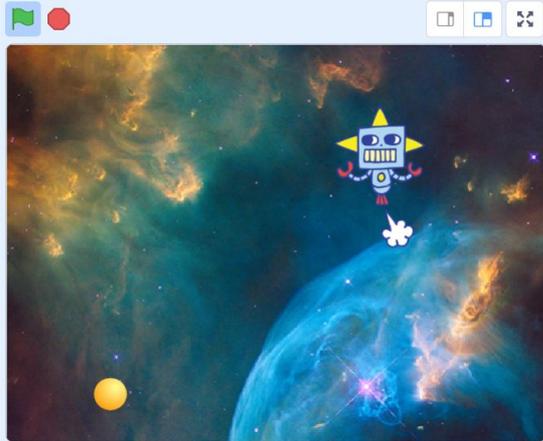
Bewege den Roboter mit den Pfeiltasten:

1. Prüfe, mit Hilfe von bedingten Anweisungen (**falls...**, **dann...**), ob eine bestimmte **Taste (z. B. Pfeil nach links) gedrückt** wird.
2. **Drehe den Roboter**, sodass er nach links guckt, wenn er sich nach links bewegt und nach rechts guckt, wenn er sich nach rechts bewegt.
3. Lass den Roboter **vom Rand "abprallen"**, damit er nicht aus dem Bild fliegt.



### Nächste Aufgabe

Wenn er vom Roboter berührt wird, soll der Ball verschwinden und zufällig an einer anderen Stelle wieder erscheinen.



### Aufgabe

Programmiere den Ball:

1. Bei **Spielstart** soll der Ball an einer **zufälligen** Position im Bild **auftauchen**.
2. **Falls** der **Roboter den Ball berührt**, lass den Ball zu einer **zufälligen** Position im Bild **gehen**.

```
Wenn Flagge angeklickt wird
  gehe zu x: Zufallszahl von -225 bis 225 y: Zufallszahl von -165 bis 165
  wiederhole fortlaufend
    falls wird Robot berührt? dann
      gehe zu x: Zufallszahl von -225 bis 225 y: Zufallszahl von -165 bis 165
```

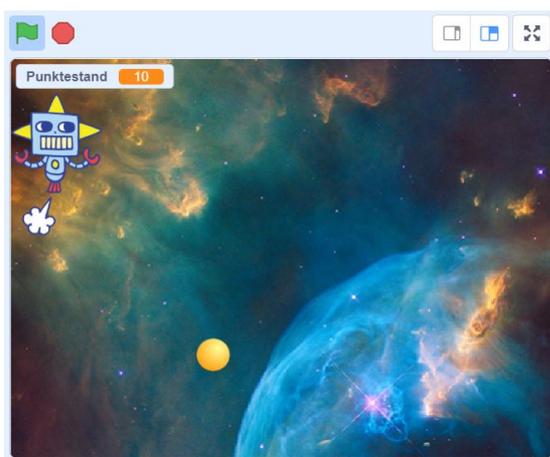


### Tipp

Hier kannst Du auch noch einen **Klang** hinzufügen, der gespielt wird, wenn der Roboter den Ball berührt.

### Nächste Aufgabe

Programmiere einen **Punktestand**. Jedes Mal, wenn den Roboter den Ball berührt, soll man einen Punkt bekommen.  
**Tipp:** Definiere dafür eine neue Variable.



### Aufgabe

Programmiere einen Punktestand:

1. **Sende** die Nachricht **“Berührt”** an alle , **wenn** der Roboter den Ball **berührt**.
2. Erstelle eine neue **Variable “Punktestand”** und **setze** den Punktestand **bei Spielstart** auf 0.
3. **Erhöhe** den Punktestand **um 1**, **wenn** die Nachricht **“Berührt” empfangen** wird.

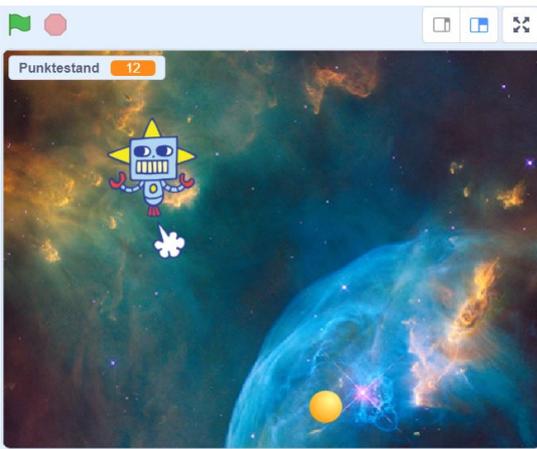
```
Wenn  angeklickt wird  
setze Punktestand auf 0
```

```
Wenn ich Berührt empfangen  
ändere Punktestand um 1
```

```
Wenn  angeklickt wird  
gehe zu x: Zufallszahl von -225 bis 225 y: Zufallszahl von -165 bis 165  
wiederhole fortlaufend  
falls wird Robot berührt? , dann  
sende Berührt an alle  
gehe zu x: Zufallszahl von -225 bis 225 y: Zufallszahl von -165 bis 165
```

### Nächste Aufgabe

Immer, wenn 2 Sekunden vergangen sind und der Ball nicht berührt wurde, soll er verschwinden und woanders im Bild wieder auftauchen.  
**Tip:** Benutze hierfür die Stoppuhr (Kategorie: Fühlen).



### Tipps

Du kannst auch Kommazahlen einsetzen, zum Beispiel 1,5 Sekunden.

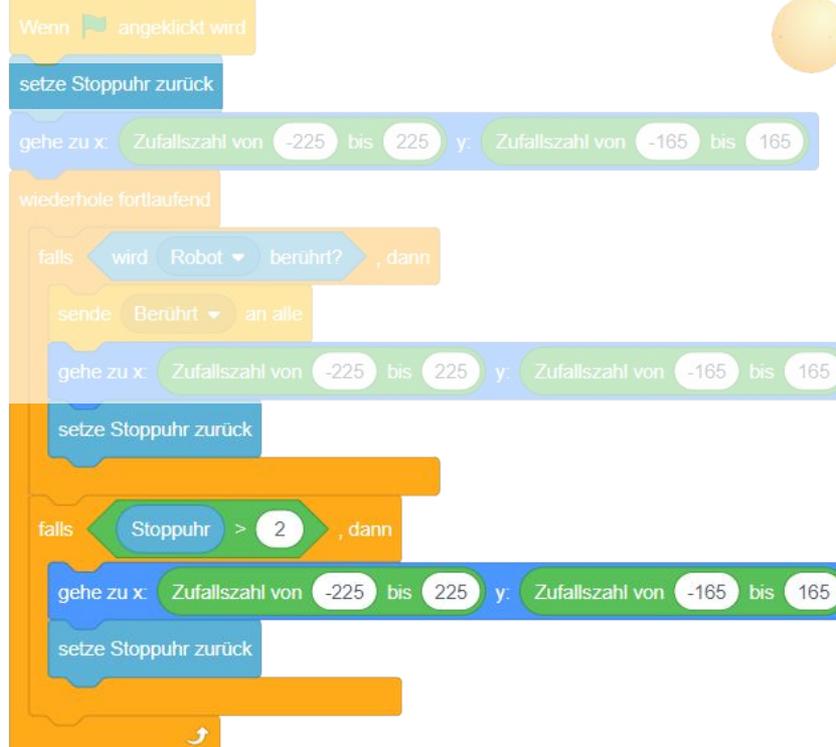
**Aber Achtung!** In Scratch musst du dann dafür 1.5 schreiben. Also mit **Punkt statt Komma!**

Wenn du die Stoppuhr schon nach einer Sekunde zurücksetzt, wird das Spiel schwieriger. Umso später, umso leichter.

### Aufgabe

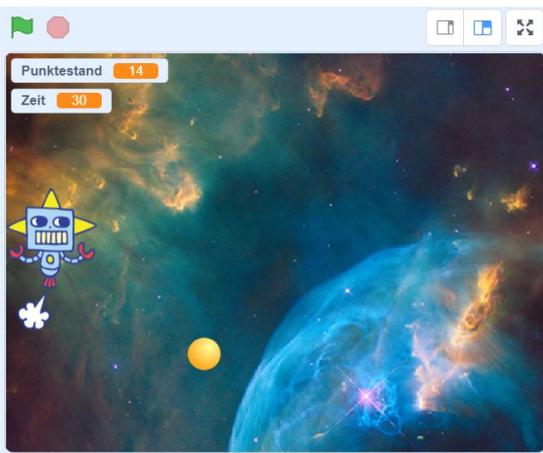
Programmiere eine Stoppuhr:

1. **Wenn** die Stoppuhr **weiter als 2 Sekunden** gezählt hat,
  - a. **lasse** den Ball zu einer **zufälligen** Position **gehen**
  - b. und **setze** die Stoppuhr wieder **zurück**.
2. **Setze** die Stoppuhr auch **bei Spielstart zurück** und, **sobald** der Roboter den Ball **berührt**.



### Nächste Aufgabe

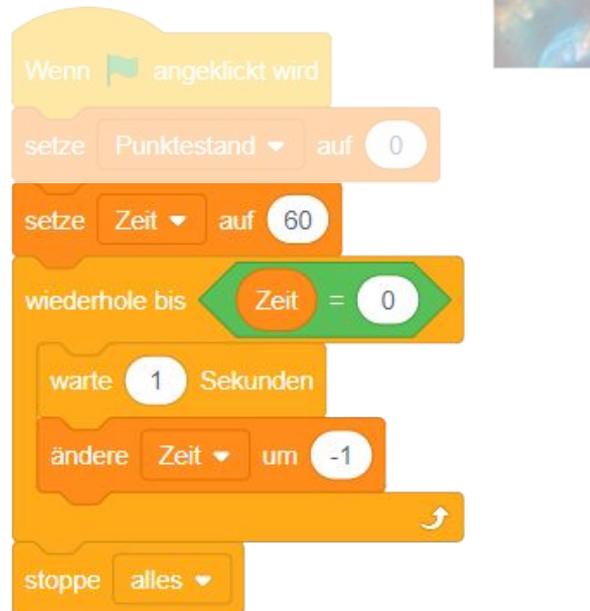
Programmiere einen Countdown. Jedes Spiel soll nur eine Minute dauern. Wenn die Zeit abgelaufen ist, soll das Spiel gestoppt werden.  **Tipp:** Für den Countdown benötigst du eine weitere Variable.



### Aufgabe

Programmiere einen Countdown:

1. Klicke rechts unten auf das **Bühnenbild** und definiere eine neue **Variable "Zeit"**.
2. **Setze** die Zeit **bei Spielstart auf 60**.
3. **Warte eine Sekunde** und **ändere** dann die Zeit **um -1**.
4. **Wiederhole** Schritt 3, **bis** die Zeit abgelaufen ist (**Zeit = 0**).
5. **Stoppe** danach das gesamte Spiel.



### Nächste Aufgabe

**Das ist schon ein tolles Spiel!**

Wenn du möchtest, kannst du das Spiel jetzt noch erweitern.

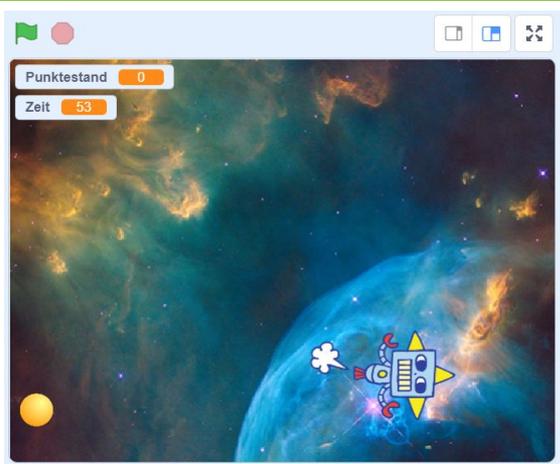
Zum Beispiel so:

- Lasse den Roboter mit dem Kopf voran fliegen.
- Sorge dafür, dass der Roboter auch schräg fliegen kann.
- Baue Buttons ein, um das Spiel zu starten.

**Auf den nächsten Lernkarten findest du Schritt für Schritt Anleitungen. Viel Spaß!**



FORTGESCHRITTEN



## Aufgabe

1. Setze beim Roboter den **Drehtyp** auf "rundherum".
2. Passe beim Fliegen nach rechts, links oben und unten die **Richtung** an, sodass der Roboter mit dem Kopf voraus fliegt (0° nach links, 180° nach rechts, usw.).

```
Wenn angeklickt wird
  gehe zu x: 0 y: -100
  setze Drehtyp auf rundherum
  wiederhole fortlaufend
    falls Taste Pfeil nach links gedrückt? dann
      setze Richtung auf 0 Grad
      ändere x um -5
    falls Taste Pfeil nach rechts gedrückt? dann
      setze Richtung auf 180 Grad
      ändere x um 5
```



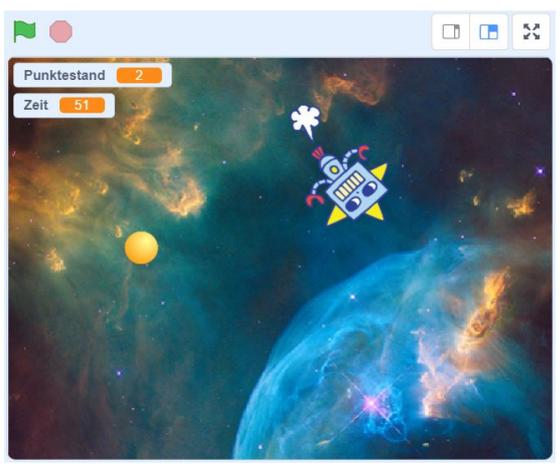
```
falls Taste Pfeil nach oben gedrückt? dann
  setze Richtung auf 90 Grad
  ändere y um 5
falls Taste Pfeil nach unten gedrückt? dann
  setze Richtung auf -90 Grad
  ändere y um -5
  pralle vom Rand ab
```

## Nächste Aufgabe

Lass den Roboter diagonal fliegen, wenn zwei Tasten gleichzeitig gedrückt werden.



FORTGESCHRITTEN



### Aufgabe

Lass den Roboter diagonal fliegen:

1. Falls die Pfeiltasten "Links" UND "Oben" gleichzeitig gedrückt sind, soll der Roboter schräg nach links oben fliegen.
2. Füge die Abfrage auch für alle übrigen diagonalen Richtungen hinzu! Überprüfe dafür, ob zwei Pfeiltasten gleichzeitig gedrückt werden. (links-unten, rechts-unten, rechts-oben)

```
falls Taste Pfeil nach links gedrückt? , dann
  setze Richtung auf 0 Grad
  ändere x um -5

falls Taste Pfeil nach links gedrückt? und Taste Pfeil nach oben gedrückt? , dann
  setze Richtung auf 45 Grad

falls Taste Pfeil nach links gedrückt? und Taste Pfeil nach unten gedrückt? , dann
  setze Richtung auf -45 Grad

falls Taste Pfeil nach rechts gedrückt? und Taste Pfeil nach unten gedrückt? , dann
  setze Richtung auf -135 Grad

falls Taste Pfeil nach rechts gedrückt? und Taste Pfeil nach oben gedrückt? , dann
  setze Richtung auf 135 Grad

pralle vom Rand ab
```



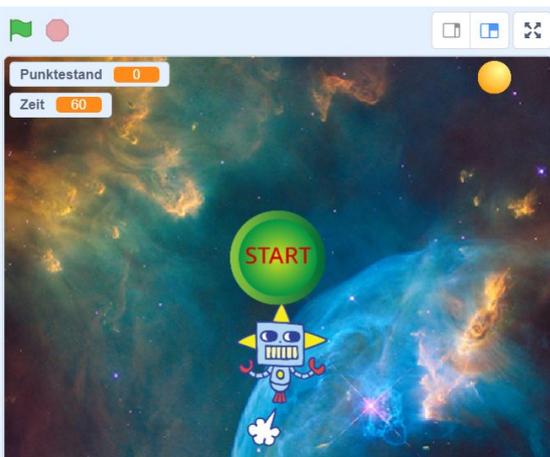
### Nächste Aufgabe

Baue einen Startknopf ein. Wenn der Knopf gedrückt wird, soll das Spiel nach 3 Sekunden starten.  
**Tipp:** Hierfür brauchst du Nachrichten, die an alle gesendet werden und Kostüme, um das Aussehen des Buttons mit jeder Sekunde zu ändern.





FORTGESCHRITTEN



Du brauchst diese Komponenten:

1 Figur Button1 ("Button")



## Tipps

Um ein Kostüm zu duplizieren, mache einen Rechtsklick darauf und klicke auf **"Duplizieren"**. Das neue Kostüm kannst du dann beliebig anpassen.



Du kannst ein Kostüm oben links umbenennen:



## Aufgabe

1. Füge einen **Button** als neue Figur hinzu.
2. Dupliziere das Kostüm vier mal und schreibe jeweils folgendes auf den Button: "START", "1", "2", "3" und "Go".
3. Benenne die Kostüme so, dass du weißt, welcher Text darauf zu sehen ist. (z.B. Start, 1, ...)
4. **Wechsle bei Spielstart zum Kostüm "Start"** und **zeige** den Button.
5. **Wenn** der Button **angeklickt** wird, **ändere das Kostüm zu "3"** und **warte eine Sekunde**. **Ändere** es dann **zu "2"** und so weiter. Zeige zuletzt das **Kostüm "Go"** und **verstecke den Button nach einer halben Sekunde**. So entsteht ein Countdown.



Wenn diese Figur angeklickt wird



Wenn  angeklickt wird

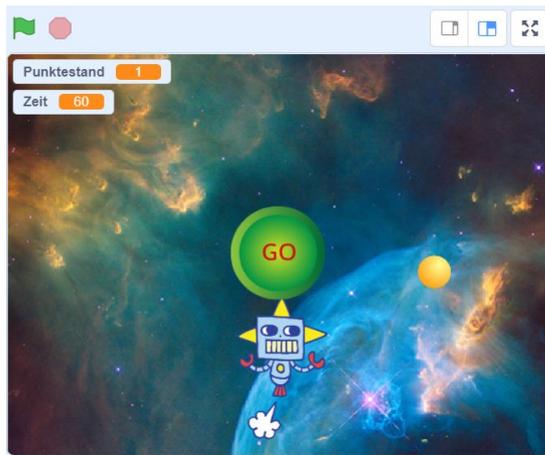


## Nächste Aufgabe

Lass das Spiel erst starten, wenn der Button verschwindet.  
**Tip:** Der Button muss eine Nachricht "Go" an alle anderen Komponenten senden. Wenn alle diese Nachricht erhalten (Countdown, Punkte, Ball...), beginnt das Spiel (nicht wie vorher nach Klick auf die grüne Flagge).

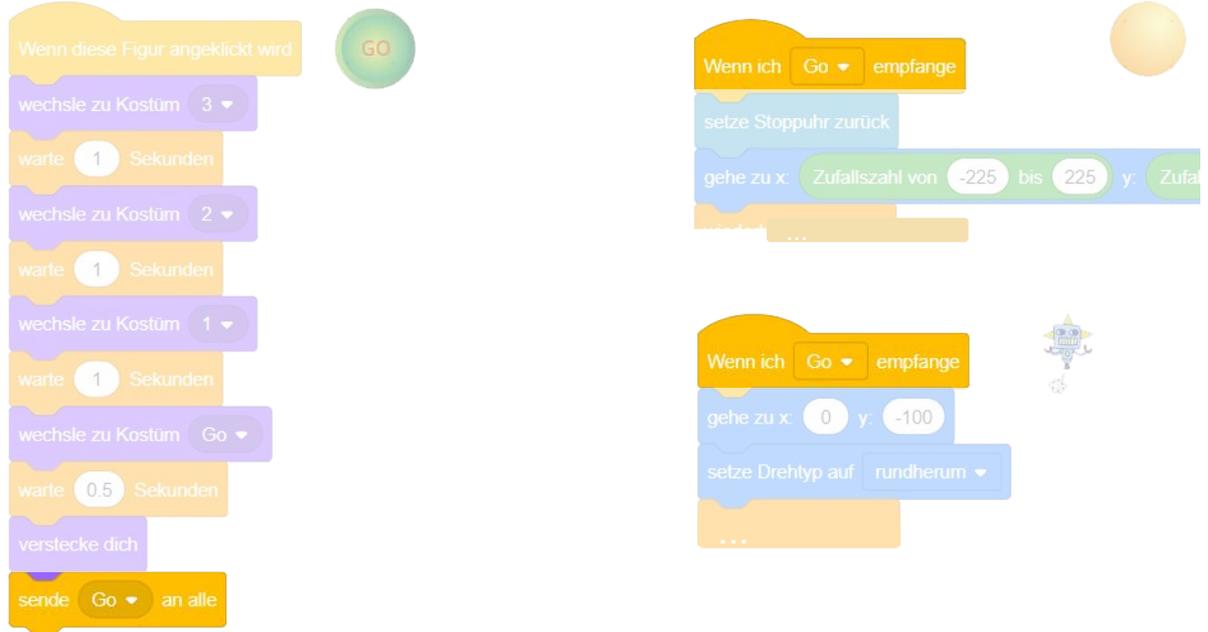


FORTGESCHRITTEN



### Aufgabe

1. Sobald der Button zum Kostüm "Go" gewechselt hat und sich versteckt, **sende eine Nachricht "Go" an alle**.
2. Ändere bei allen Komponenten (außer dem Button):  
"Wenn angeklickt wird" zu "Wenn ich 'Go' empfange".



### Tipp

Füge einen Klang hinzu, jedes Mal, wenn die Zahl umspringt. Dadurch wirkt der Countdown noch authentischer.

### Nächste Aufgabe

Du hast ein tolles Spiel programmiert. **Gratuliere!** Wenn du möchtest, erweitere dein Spiel beliebig. Lass deiner Kreativität freien Lauf.