



Projekt starten

- Starte ein neues Projekt:

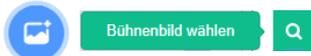


- Gib dem Projekt den Namen *Verstecken und finden*.

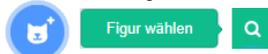


Sich zeigen und verstecken

- Wähle als passenden Hintergrund *Mountain*.



- Lösche die Katzen-Figur mit Rechtsklick und wähle stattdessen die Figur *Gobo*.



- Mit dem folgenden Skript erreichst du, dass Gobo sich jeweils für eine Sekunde versteckt und sich dann wieder für eine Sekunde zeigt.

Diese Zahl gibt an, wie lange Gobo sich versteckt.

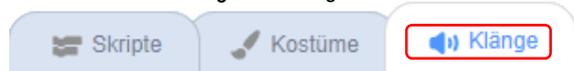
Diese Zahl gibt an, wie lange Gobo sich zeigt.

- Teste das Skript. Macht es, was es sollte?



Gobo sprechen lassen

- Wenn Gobo spricht, soll vorher noch ein passender Klang ertönen. Du kannst ihm einen neuen Klang zuordnen. Wechsle dafür ins Register *Klänge*.



- Geh auf die Suche nach dem passenden Klang. (Beispielsweise: *hey*) und wähle ihn aus.



- Wechsle ins Register *Skripte* und schreibe folgendes Skript:

Setze hier deinen Text ein.

Wähle hier den gewünschten Klang aus.

- Teste dein Skript, indem du zuerst auf die grüne Flagge klickst und danach auf *Gobo*.



Überraschung

- *Gobo* kann ein verschiedenes Aussehen haben. Wechsle ins Register *Kostüme*. Wähle das dritte Kostüm, indem du darauf klickst (*Gobos* Aussehen hat sich nun geändert).
- *Gobo* soll nun nicht nach jeweils einer Sekunde erscheinen, sondern nach einer zufällig gewählten Zeit. Wechsle dafür zurück ins *Skripte*-Register und ändere das erste Skript folgendermaßen ab:



Der Zufallsblock gibt eine zufällige Zahl aus dem vorgegebenen Bereich an das Skript weiter.

- Teste das Skript und probiere verschiedene Zahlen im Zufallsblock aus.



An einem zufälligen Ort auftauchen

Gobo kann nicht nur zu einer zufälligen Zeit, sondern auch an einem zufälligen Ort auftauchen.

- Ändere das Skript weiter ab:

- Teste das Skript. *Gobo* sollte nun nach dem Klicken auf die grüne Flagge an verschiedenen Orten auftauchen.

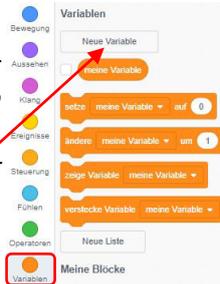




Punkte zählen

Als Nächstes baust du in dein Projekt einen Zähler ein, der dir sagt, wie oft du auf Gobo geklickt hast.

- Wähle in der Block-Palette **Variablen**.
- Klicke auf **Neue Variable**.



- Nenne die neu erstellte Variable im auftauchenden Feld **Punkttestand** und klicke danach auf **OK**.



- Neu siehst du nun auf der Bühne ein oranges Feld, das den Punkttestand angibt. Du kannst es dorthin verschieben, wo es dir am passendsten erscheint.



- Schreibe ein neues Skript, welches den Punkttestand auf null setzt, wenn du die grüne Flagge klickst:



- Ergänze das **bereits vorhandenes** Skript so, dass der Punkttestand jeweils um eins erhöht wird, wenn auf Gobo geklickt wird.

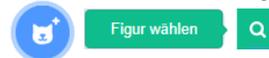


- Teste deine Skripte und hole möglichst viele Punkte! 🎮

Ein Platz zum Verstecken

Gobo soll sich zu Beginn dieses Spiels hinter einem Stein versteckt halten.

- Wähle eine zusätzliche Figur: **Rocks**

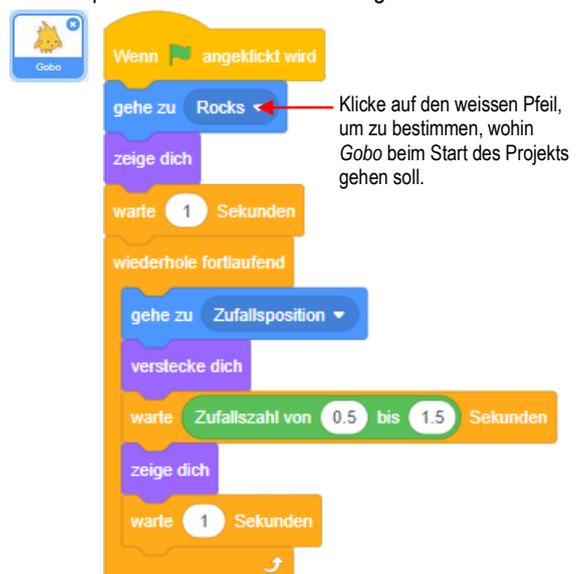


- Schreibe folgendes Skript für die Steine:



Dieser Block bewirkt, dass die Figur **Rocks** ganz zuvorderst auf der Bühne hinkommt. Mit anderen Worten, sie wird **Gobo** verdecken.

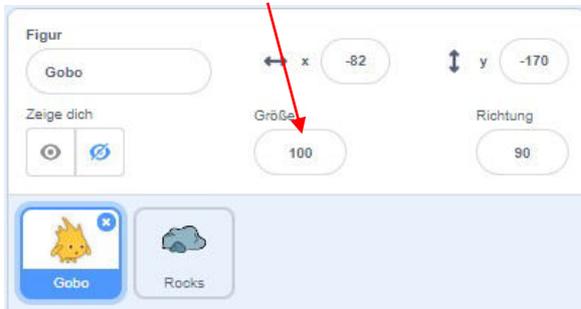
- Das Skript von **Gobo** erweiterst du folgendermassen:



Klicke auf den weissen Pfeil, um zu bestimmen, wohin Gobo beim Start des Projekts gehen soll.



Die Grösse von Gobo oder den Steinen kannst du ändern, indem du die entsprechende Figur auswählst und die Zahl für die Grösse bei den Figuren-Infos änderst:



- Teste die Skripts. Gobo sollte nach dem Klick auf die grüne Flagge nun hinter den Steinen starten. 

Jetzt bist du dran!

Hast du eine gute Idee, wie du das jetzt Gelernte in einem Scratch-Projekt umsetzen könntest? Dann programmiere sie!

Ansonsten kannst du folgende Vorschläge ausprobieren:

- Erstelle ein neues Projekt: Programmiere Gobo so, dass er an einer zufälligen Stelle startet. Jedes Mal, wenn er angeklickt wird, soll er einen neuen, zufälligen Standort einnehmen, ein Geräusch von sich geben und „Du hast mich entdeckt“ für zwei Sekunden sagen. Gleichzeitig soll beim Punktezähler ein Punkt dazugezählt werden. Damit das Finden von Gobo nicht so einfach ist, soll er die ganze Zeit unsichtbar bleiben. Benütze dazu folgenden Block:



- Erstelle ein neues Projekt: Komponiere eine Landschaft mit verschiedenen Figuren, welche zum Hintergrund passen. Hinter einer dieser Figuren ist der verkleinerte Gobo versteckt. Der Anwender soll nun Gobo suchen, indem er auf die verschiedenen Figuren klickt. Diese verschwinden dann für eine Sekunde, so dass man sehen kann, ob Gobo dahinter versteckt war. Das Projekt soll zählen, wie viele Figuren angeklickt werden mussten, bis Gobo gefunden wurde.
- Erstelle ein neues Projekt: Verteile verschiedene Figuren deiner Wahl auf der Bühne. Programmiere diese so, dass sie verschwinden, wenn sie von der fliegenden Katze (Cat Flying) berührt werden. Gleichzeitig soll bei einem Punktezähler ein Punkt dazugezählt werden. Die fliegende Katze programmierst du so, dass sie mit den Pfeiltasten gesteuert werden kann.

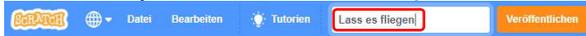


Projekt starten

- Starte ein neues Projekt:

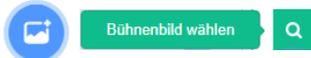


- Gib dem Projekt den Namen *Lass es fliegen*.



Eine fliegende Figur auswählen

- Wähle als passenden Hintergrund *Blue Sky 2*.



- Lösche die Katzen-Figur mit Rechtsklick und wähle stattdessen die Figur *Cat Flying*.



- Schreibe folgendes Skript für die fliegende Katze:

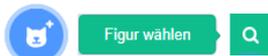


Lässt die Katze zuvorderst und alle anderen Figuren bleiben im Hintergrund.

- Teste das Skript. Macht es, was es sollte?

Mit Fliegen beginnen

- Wähle eine zusätzliche Figur: *Buildings*



- Ziehe die Figur auf der Bühne so nach unten, dass zwischen Figur und Bühnenrand kein freier Platz bleibt.



- Schreibe das folgende Skript, welches das Gebäude an der Katze vorbeiziehen lässt:



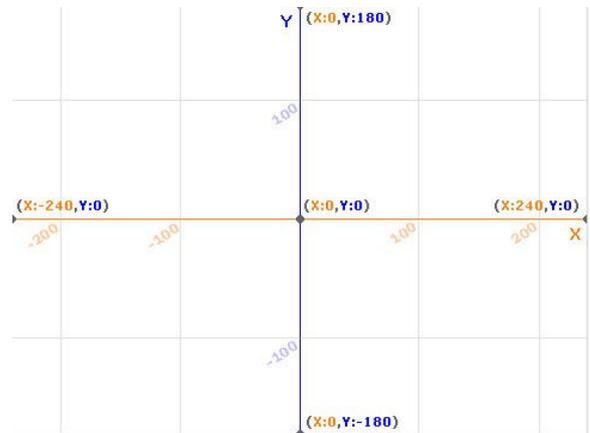
Setzt das Gebäude etwas über den rechten Rand hinaus.

Eine negative Zahl verschiebt das Gebäude von rechts nach links.

Beachte:

X bestimmt, wie weit links oder rechts eine Figur auf der Bühne zu stehen kommt. -240 ist ganz links und 240 ist ganz rechts. 0 ist in der Mitte.

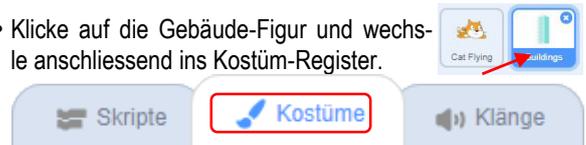
Y bestimmt, wie weit oben oder unten eine Figur auf der Bühne zu stehen kommt. -180 ist ganz unten und 180 ist ganz oben. 0 ist in der Mitte.



- Teste dein Skript. Nach dem Klick auf die grüne Flagge sollte die Katze am Wolkenkratzer vorbeifliegen.

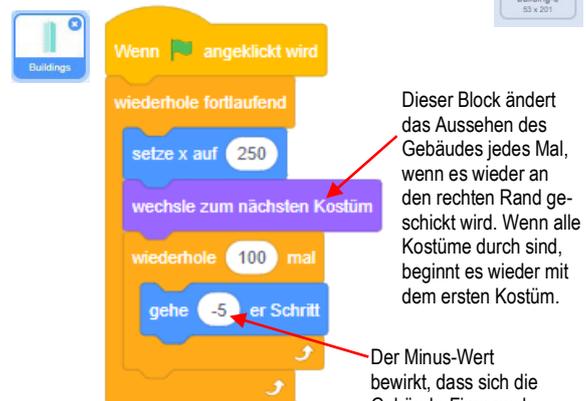
Das Aussehen ändern

- Klicke auf die Gebäude-Figur und wechsle anschließend ins Kostüm-Register.



Du entdeckst, dass die Figur *Buildings* verschieden aussehen kann, da sie insgesamt zehn verschiedene Kostüme hat.

- Wechsle nun zurück ins Skripte-Register und verändere das Gebäude-Skript so, dass das Aussehen des Gebäudes sich jedes Mal ändert, wenn das Gebäude am rechten Rand zu stehen kommt:



Dieser Block ändert das Aussehen des Gebäudes jedes Mal, wenn es wieder an den rechten Rand geschickt wird. Wenn alle Kostüme durch sind, beginnt es wieder mit dem ersten Kostüm.

Der Minus-Wert bewirkt, dass sich die Gebäude-Figur nach links bewegt.

- Teste das Skript: Fliegt die Katze an zehn verschiedenen Gebäuden vorbei, nachdem du die grüne Flagge geklickt hast?





Die Katze steuern

- Schreibe für die Katzen-Figur folgende Skripts, damit du sie mit den Pfeiltasten steuern kannst:

```

Wenn Taste Pfeil nach links gedrückt wird
  ändere x um -10

Wenn Taste Pfeil nach rechts gedrückt wird
  ändere x um 10

Wenn Taste Pfeil nach oben gedrückt wird
  ändere y um 10

Wenn Taste Pfeil nach unten gedrückt wird
  ändere y um -10
  
```

- Teste die Skripts. Kannst du die Katze nun mit den Pfeiltasten steuern?

Vorbeifliegende Wolken

- Wähle eine neue Figur dazu: *Clouds*
- Schreibe folgendes Skript dazu:

```

Wenn angeklickt wird
  wiederhole fortlaufend
    setze x auf 250
    setze y auf Zufallszahl von 1 bis 180
    wechsle zum nächsten Kostüm
  wiederhole 50 mal
  ändere x um -10
  
```

Lässt die Wolken am rechten Rand starten.

Legt den Start der Wolke irgendwo in der oberen Bühnenhälfte fest. (Für y bedeutet 0 die Mitte und 180 die obere Rand und -180 die untere Rand.)

Wechselt für jede neue Wolke das Aussehen

- Teste das Skript. Fliegen verschiedene Wolken in verschiedenen Höhen an der Katze vorbei, wenn du die grüne Flagge geklickt hast?

Fliegende Herzen

- Wähle eine zusätzliche Figur: *Heart*
- Schreibe folgendes Skript dazu:

```

Wenn angeklickt wird
  wiederhole fortlaufend
    setze Größe auf 50
    gehe zu Zufallsposition
    setze x auf 250
    wiederhole 32 mal
      ändere x um -15
  
```

Verkleinert die Größe des Herzens auf die Hälfte.

Wählt eine zufällige Startposition für das Herz aus

Verschiebt die zufällige Position an den rechten Rand.

Bestimmt die Geschwindigkeit des Herzens.

- Teste das Skript. Fliegen der Katze nun Herzen entgegen, wenn du die grüne Flagge klickst?

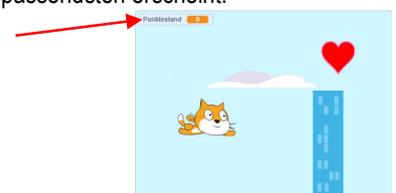
Punkte sammeln

Als Letztes baust du in dein Projekt einen Zähler ein, der zählt, wie viele Herzen du mit der fliegenden Katze berühren kannst.

- Wähle in der Block-Palette *Variablen*.
- Klicke auf *Neue Variable*.
- Nenne die neu erstellte Variable im auftauchenden Feld *Punkttestand* und klicke danach auf OK.



- Neu siehst du nun auf der Bühne ein oranges Feld, das den Punkttestand angibt. Du kannst es dorthin verschieben, wo es dir am passendsten erscheint.





- Schreibe folgendes zusätzliches Katzen-Skript um den Zähler in Betrieb zu nehmen:

Setzt den Anfangspunktestand auf 0.

Zählt jeweils einen Punkt zum Punktestand dazu, wenn ein Herz berührt wird.

Die Schleife stellt sicher, dass die Herzberührung nicht nur bei Programmstart, sondern während des ganzen Spiels überprüft wird.

- Teste dein Skript. Funktioniert der Zähler, wenn du die grüne Flagge geklickt hast? 

Jetzt bist du dran!

Hast du eine gute Idee, wie du das jetzt Gelernte in einem Scratch-Projekt umsetzen könntest? Dann programmiere sie!

Ansonsten kannst du folgende Vorschläge ausprobieren:

- Kannst du dieses Projekt so ergänzen, dass vom Punktestand jedes Mal ein Punkt abgezählt wird, wenn die Katze ein Gebäude berührt?
- Kannst du dieses Projekt so ergänzen, dass jedes Mal ein Klang ertönt, wenn die Katze ein Herz erwischt und das Herz danach verschwindet (Achte aber darauf dass es danach am rechten Bildrand wieder auftaucht!)
- Kannst du ein eigenes, neues Projekt schreiben, bei dem eine Figur anderen Figuren, die von rechts kommen, ausweichen muss. Bei Berührung gibt es einen Minuspunkt.



Projekt starten

- Starte ein neues Projekt:

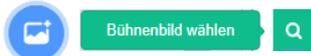


- Gib dem Projekt den Namen *Fangspiel*.

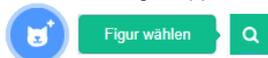


Einen Apfel an den oberen Rand setzen

- Wähle als passenden Hintergrund *Blue Sky*.



- Lösche die Katzen-Figur mit Rechtsklick und wähle stattdessen die Figur *Apple*.



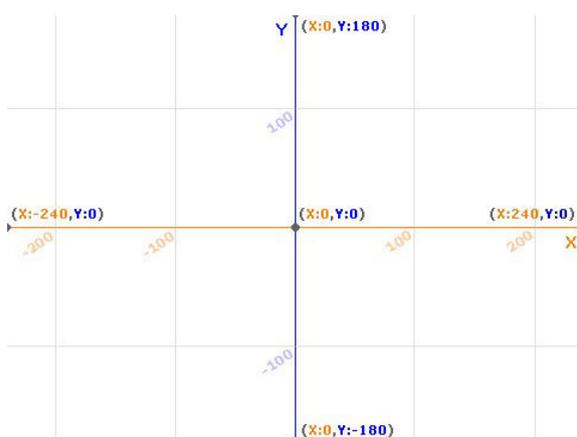
- Der Apfel soll zu Beginn des Spieles immer am oberen Rand erscheinen. Schreibe deshalb folgendes Skript:



Position:

Y bestimmt, wie weit oben oder unten eine Figur auf der Bühne zu stehen kommt. -180 ist ganz unten und 180 ist ganz oben. 0 ist in der Mitte.

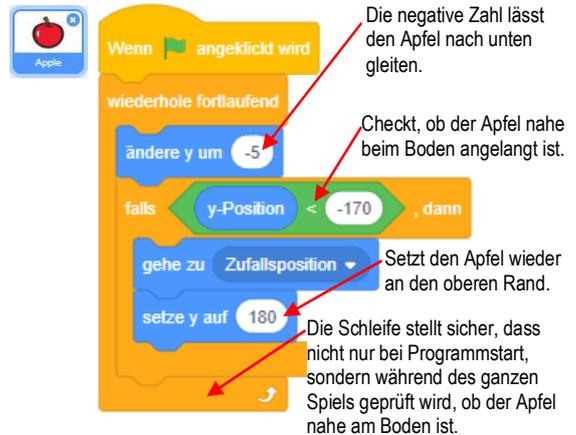
X bestimmt, wie weit links oder rechts eine Figur auf der Bühne zu stehen kommt. -240 ist ganz links und 240 ist ganz rechts. 0 ist in der Mitte.



- Teste das Skript. Wenn du ein paar Mal auf die grüne Flagge klickst, sollte der Apfel jedes Mal an einem anderen Ort am oberen Rand erscheinen.

Runterfallen

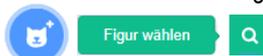
- Um den Apfel langsam zu Boden fallen zu lassen schreibe folgendes Skript:



- Teste das Skript. Es sollte laufen ein Apfel runterfallen.

Einen Fänger bereitstellen

- Wähle eine zusätzliche Figur: *Bowl*



- Ziehe die Schale auf den Boden der Bühne (brauner Bereich).

- Um den Fänger hin und her zu bewegen schreibe folgendes Skript:



- Ebenfalls möglich wäre die Tastatursteuerung mit den nebenstehenden Skripts. Probiere beide, die obere und die nebenstehende Variante nacheinander aus, indem du den Fänger mit der linken und rechten Pfeiltaste steuerst und entscheide dich danach für die Variante, welche besser läuft.



Fang ihn!

Wenn ein Apfel auf die Schale fällt, soll dies mit einem Ton angezeigt werden und der Apfel soll ein weiteres Mal vom Himmel fallen.

- Klicke auf die Apfel-Figur und schreibe folgendes Skript:



Wenn **angeklickt wird**

wiederhole fortlaufend

falls **wird Bowl berührt?** , dann

spiele Klang **Chomp**

gehe zu **Zufallsposition**

setze y auf **180**

• Teste das Skript.

Wähle die *Bowl* mit dem kleinen weißen Pfeil aus.

Die Schleife stellt sicher, dass nicht nur bei Programmstart, sondern während des ganzen Spiels geprüft wird, ob die Figur *Bowl* berührt wird.

Den Klang ändern

Probiere das Skript auch einmal mit einem anderen Klang von der Klangbibliothek aus:

- Wechsle ins Klangregister.
- Geh auf die Suche nach einem passenden Klang.



- Schränke dafür die Auswahl ein, indem du auf *Effekte* klickst.



- Hast du den Klang gefunden, klicke auf ihn (Beispielsweise: *zoop*). In der Klänge Liste erscheint der neue Klang:



- Wechsle ins Skripte-Register.
- Ändere das bestehende Skript:

Wenn **angeklickt wird**

wiederhole fortlaufend

falls **wird Bowl berührt?** , dann

spiele Klang **Zoop**

gehe zu **Zufallsposition**

setze y auf **180**

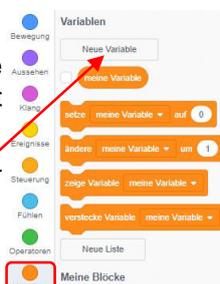
• Teste das Skript.

Wähle hier den neuzugeordneten Klang aus.

Zähle die Punkte

Sicher willst du wissen, wie viele Äpfel du gefangen hast. Das lässt sich einrichten.

- Wähle in der Block-Palette *Variablen*.
- Klicke auf *Neue Variable*.



- Nenne die neu erstellte Variable im auftauchenden Feld *Punktstand* und klicke danach auf *OK*.



- Neu siehst du nun auf der Bühne ein oranges Feld, das den Punktstand angibt. Du kannst es dorthin verschieben, wo es dir am passendsten erscheint.



- Füge dem bestehenden Apfel-Skript zwei neue Blocks zu:

Wenn **angeklickt wird**

setze **Punktstand** auf **0**

wiederhole fortlaufend

falls **wird Bowl berührt?** , dann

spiele Klang **Zoop**

ändere **Punktstand** um **1**

gehe zu **Zufallsposition**

setze y auf **180**

Dieser Block setzt den Punktstand zu Beginn des Spiels auf 0.

Dieser Block zählt jedes Mal einen Punkt zum Punktstand dazu, wenn ein Apfel gefangen wurde.

- Teste das Skript. Funktioniert der Zähler?



Bonus-Punkte

Goldene Äpfel geben die doppelte Punktzahl!

- Dupliziere die Apfel-Figur zusammen mit allen bisher geschriebenen Skripts durch einen Rechtsklick auf die Figur.
- Klicke auf die *Apple2*-Figur und anschliessend auf das *Kostüme*-Register.
- Wähle den Fülleimer und danach die Füllfarbe. Mixe mit den Schiebreglern eine gelbe Farbe zusammen.



- Wenn du nun auf den Apfel klickst, wird der gelb.
- Wechsle in das *Skripte*-Register.
- Ändere das vorhandene Skript so ab, dass der Zähler jedes Mal zwei Punkte zählt, wenn ein gelber Apfel gefangen wird:

```

Wenn angeklickt wird
  setze Punktestand auf 0
  wiederhole fortlaufend
    falls wird Bowl berührt? , dann
      spiele Klang Zoop
      ändere Punktestand um 2
      gehe zu Zufallsposition
      setze y auf 180
  
```

- Teste das Skript. Unterscheidet der Zähler zwischen roten und gelben (goldenen) Äpfel?

Den Gewinn anzeigen

- Wähle eine neue, leere Figur, indem du *Malen* auswählst.
- Gib der Figur den Namen *Schluss-Anzeige*.



- Klicke auf das Textwerkzeug **T** und schreibe *Gewonnen!*



- Du kannst bei diesem Schriftzug...

- die Farbe ändern:
- die Schriftart ändern:
- die Grösse ändern: Klicke dafür auf das *Auswählen*-Werkzeug und ziehe danach an einem der erscheinenden blauen Eckpunkte.



- Wechsle ins *Skripte*-Register.
- Schreibe ein Skript zur Text-Figur, welches den Gewinn des Spieles anzeigen lässt.

```

Wenn angeklickt wird
  verstecke dich
  warte bis Punktestand > 5
  zeige dich
  stoppe alles

```

Bedeutet, dass der Punktestand grösser als 5 sein muss, um zu gewinnen.

- Teste das Skript, indem du solange spielst, bis du über die 5 Punkte gekommen bist. Natürlich kannst du auch eine andere Punktezahl vorgeben. Probiere es aus.

Jetzt bist du dran!

Hast du eine gute Idee, wie du das jetzt Gelernte in einem Scratch-Projekt umsetzen könntest? Dann programmiere sie!

Ansonsten kannst du folgende Vorschläge ausprobieren:

- Lass in diesem Projekt die roten Äpfel langsamer runterfallen und die gelben schneller.
- Ergänze dieses Projekt noch mit braunen Äpfeln, welche 3 Punkte Abzug geben.
- Erstelle ein neues Projekt: Programmiere die Katze so, dass sie mit den Pfeiltasten über die Bühne gesteuert werden kann. Jedes Mal, wenn sie in die Nähe eines Bühnenrandes kommt, soll die Figur durch Miauen warnen. Nachfolgend siehst du als Beispiel das Skript für den oberen Rand. Programmiere die anderen drei Skripte.

```

Wenn Taste Pfeil nach oben gedrückt wird
  ändere y um 5
  falls y-Position > 120 , dann
    spiele Klang Miau
  
```



Projekt starten

- Starte ein neues Projekt:



- Gib dem Projekt den Namen *Pong*.



Den Ball springen lassen

- Wähle als passenden Hintergrund *Neon Tunnel*.



- Lösche die Katzen-Figur mit Rechtsklick und wähle stattdessen die Figur *Ball*.



- Um den Ball springen zu lassen, schreibe folgendes Skript:

Wenn **angeklickt** wird

gehe zu x: **0** y: **160**

setze Richtung auf **45** Grad

wiederhole fortlaufend

gehe **15** er Schritt

pralle vom Rand ab

Legt fest, von welchem Ort der Ball startet.

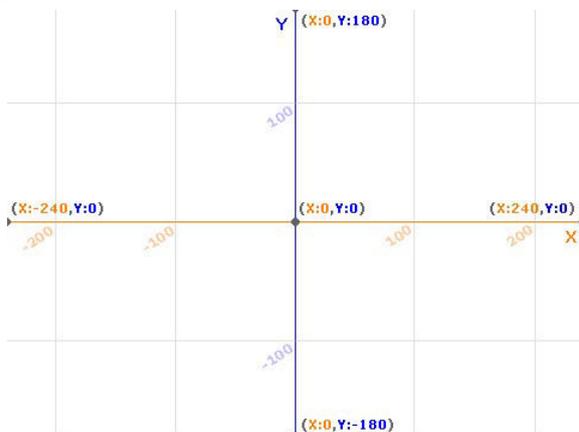
Legt fest, in welche Richtung sich der Ball am Anfang bewegt.

Hier kann die Geschwindigkeit des Balles eingegeben werden. (grössere Zahl = schneller, kleinere Zahl = langsamer)

Position:

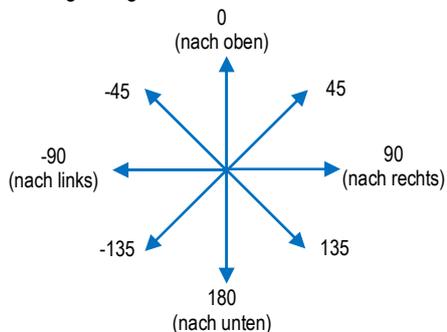
X bestimmt, wie weit links oder rechts eine Figur auf der Bühne zu stehen kommt. -240 ist ganz links und 240 ist ganz rechts. 0 ist in der Mitte.

Y bestimmt, wie weit oben oder unten eine Figur auf der Bühne zu stehen kommt. -180 ist ganz unten und 180 ist ganz oben. 0 ist in der Mitte.



Richtung:

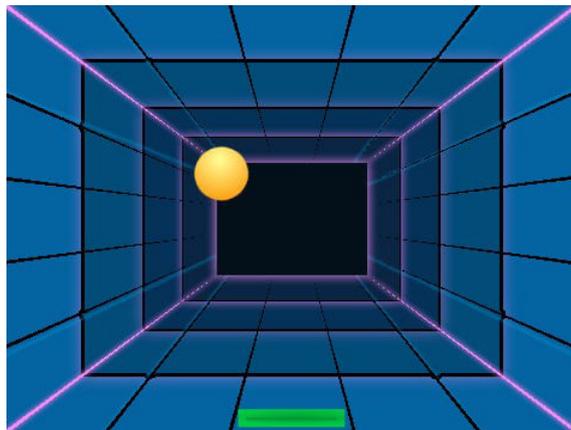
Die Richtung, in welche sich eine Figur fortbewegt, wird mit einem Winkelmass angegeben. Am nachfolgenden Beispiel kannst du einige Möglichkeiten ablesen:



- Probiere das Skript mit verschiedenen Eingaben bei den Blöcken aus und beobachte die Reaktionen.

Ein Paddel einrichten

- Wähle eine zusätzliche Figur: *Paddle*.
- Ziehe das Paddel auf der Bühne an den unteren Rand.



- Schreibe das folgende Skript, durch welches das Paddel mit der Maus gesteuert werden kann.

Wenn **angeklickt** wird

wiederhole fortlaufend

setze x auf **Maus x-Position**

Bewirkt, dass das Paddel Bewegung der Maus folgt, allerdings nur in horizontaler Richtung.

- Teste das Skript. Das Paddel sollte jetzt den Mausbewegungen folgen.

Den Ball vom Paddel abprallen lassen

- Der Ball bekommt jetzt noch ein weiteres Skript, um ihn vom Paddel abprallen zu lassen:



```

Wenn angeklickt wird
  wiederhole fortlaufend
    falls wird Paddle berührt? , dann
      drehe dich um 180 Grad
  
```

Klicke auf den kleinen, weissen Pfeil, um Paddle auszuwählen.

Die Schleife stellt sicher, dass nicht nur bei Programmstart, sondern während des ganzen Spiels überprüft wird, ob das Paddle berührt wird.

- Teste das Skript. Der Ball sollte nun vom Paddle abprallen.

Den Zufall ins Spiel bringen

- Im Moment reagiert der Ball noch sehr vorhersehbar. Er prallt immer im gleichen Winkel ab, so dass das Spiel nicht sehr herausfordernd ist. Ändere deshalb das Ballskript wie folgt ab:

```

Wenn angeklickt wird
  wiederhole fortlaufend
    falls wird Paddle berührt? , dann
      drehe dich um Zufallszahl von 170 bis 190
  
```

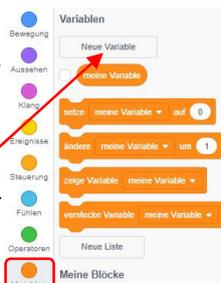
Zahlen in der Nähe von 180 lassen den Ball in leicht verschiedene Richtungen abprallen.

- Teste das Skript. Probiere beim Zufallszahl-Block auch Zahlen mit einer grösseren Abweichung von 180 aus.

Punkte zählen

Als Nächstes baust du in dein Projekt einen Zähler ein, der dir sagt, wie oft du den Ball mit dem Padel jongliert hast.

- Wähle in der Block-Palette **Variablen**.
- Klicke auf **Neue Variable**.
- Nenne die neu erstellte Variable im auftauchenden Feld **Punktstand** und klicke danach auf **OK**.



- Ergänze **das bereits bestehende** Ball-Skript so, dass bei Spielbeginn der Punktstand auf null gesetzt wird und für jede Berührung mit dem Paddle ein Punkt dazugezählt wird:

```

Wenn angeklickt wird
  setze Punktstand auf 0
  wiederhole fortlaufend
    falls wird Paddle berührt? , dann
      drehe dich um Zufallszahl von 170 bis 190
      ändere Punktstand um 1
      warte 0.5 Sekunden
  
```

- Neu siehst du nun auf der Bühne ein oranges Feld, das den Punktstand angibt. Du kannst es dorthin verschieben, wo es dir am passendsten erscheint.



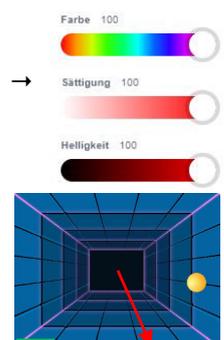
- Teste das Skript. Der Zähler sollte nun funktionieren.

Das Ende des Spiels

- Wähle das Bühnenbild durch Anklicken und wechsle danach ins Register **Bühnenbilder**.



- Klicke auf das **Linien-Werkzeug**.
- Stelle die **Linien-Dicke** auf 20 ein.
- Wähle danach die **Farbe Rot** aus.
- Zeichne eine rote Linie nahe am Boden. Halte dabei die **Shift-Taste** gedrückt, damit die Linie waagrecht ausgerichtet wird.





- Wähle danach die Ball-Figur aus.
- Wechsle ins *Skripte*-Register.
- Das dritte Ball-Skript soll bewirken, dass das Spiel verloren ist, sobald der Ball den Boden (rote Linie) berührt.

Um eine Farbe auszuwählen, klicke zuerst in dieses Feld, danach auf die Farbpipette unterhalb der Schieberegler und schliesslich mit dem auftauchenden Vergrößerungsglas auf die gewünschte Farbe auf der Bühne, hier also auf die rote Linie.

- Das letzte Ball-Skript bestimmt, dass das Spiel gewonnen ist, wenn der Ball zwanzigmal das Paddel berührt hat:

Natürlich könntest du hier auch eine andere Zahl einsetzen.

- Danach wählst du eine neue, leere Figur, indem du *Malen* auswählst.



- Gib der Figur den Namen *Schluss-Anzeige*.

- Klicke auf das Textwerkzeug **T** und schreibe *Verloren!*



- Du kannst bei diesem Schriftzug...

- die Farbe ändern:
- die Schriftart ändern:
- die Grösse ändern: Klicke dafür auf das *Auswählen*-Werkzeug und ziehe danach an einem der erscheinenden blauen Eckpunkten.



- Verändere nun den Schriftzug nach deinem Geschmack und ziehe ihn danach in die Mitte der Zeichnungsfläche, solange das *Auswählen*-Werkzeug noch angewählt ist.

- Ändere links oben den Namen des *Kostüm1* in *Verloren!*

- Dupliziere das Kostüm *Verloren!* durch Rechtsklick und ändere links oben seinen Namen in *Gewonnen!*



- Die folgenden drei Skripts lassen die Gewinn- oder Verloren-Anzeige im richtigen Moment erscheinen.

- Teste die Skripts. Bei Spielende sollten nun die passenden Anzeigen erscheinen.

Jetzt bist du dran!

Hast du eine gute Idee, wie du das jetzt Gelernte in einem Scratch-Projekt umsetzen könntest? Dann programmiere sie!

Ansonsten kannst du folgende Vorschläge ausprobieren:

- Ergänze dieses Projekt so, dass es jedes Mal, wenn der Ball vom Paddel zurückgeschlagen wird, ein passender Klang ertönt.
- Erweitere das Projekt weiter, indem du die Figur *Button2* oder *Button3* einige Male vervielfachst und diese an den oberen Rand der Bühne hinstellst. Sobald der Ball eine von den Figuren berührt, verschwindet sie. Gewonnen hat die Spielerin, wenn alle Buttons verschwunden sind.
- Untermale auch das Verschwinden der Buttons mit passenden Tönen.

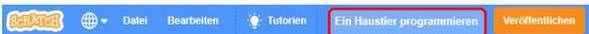


Projekt starten

- Starte ein neues Projekt:



- Gib dem Projekt den Namen *Ein Haustier programmieren*.

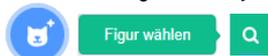


Dein Haustier stellt sich vor

- Wähle als passenden Hintergrund *Garden-rock*.



- Lösche die Katzen-Figur mit Rechtsklick und wähle stattdessen die Figur *Monkey*.

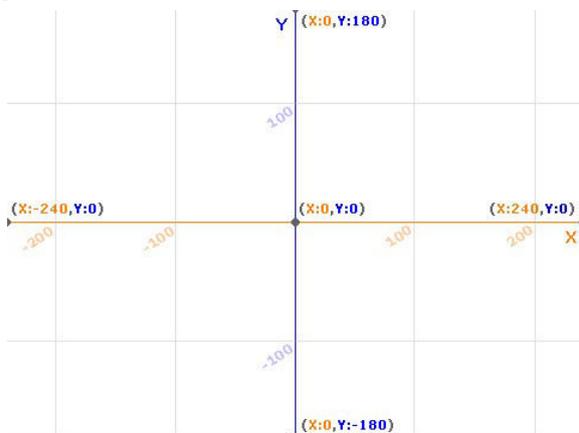


- Lass dein Haustier sich vorstellen. Schreibe dazu folgendes Skript:

Position:

X bestimmt, wie weit links oder rechts eine Figur auf der Bühne zu stehen kommt. -240 ist ganz links und 240 ist ganz rechts. 0 ist in der Mitte.

Y bestimmt, wie weit oben oder unten eine Figur auf der Bühne zu stehen kommt. -180 ist ganz unten und 180 ist ganz oben. 0 ist in der Mitte.



- Teste das Skript. Ändere dazu auch die Position von deiner Figur, indem du andere x- und y-Werte eingibst und das Äffchen verschiedene Sachen sagen lässt.

Beweg dein Haustier

- Klicke auf das Kostüme-Register.



Da die Figur *Monkey* drei verschiedene Kostüme hat, kann sie verschieden aussehen. Das kannst du dazu ausnützen, um das Äffchen zu bewegen.

- Klicke auf das Skripte-Register zurück.



- Schreibe folgendes Skript:

- Teste das Skript. Wenn du auf das Äffchen klickst, sollte es sich bewegen und sich hörbar machen.

Dein Haustier füttern

Fürs Füttern benötigen wir einen neuen Klang. Den kannst du dem Äffchen folgendermassen zuordnen:

- Wechsle ins Klangregister.



- Geh auf die Suche nach einem Klang.



- Schränke dafür die Auswahl ein, indem du auf *Effekte* klickst.



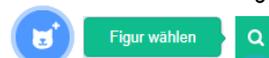
- Hast du einen passenden Klang gefunden, klicke auf ihn (Beispielsweise: *Chomp*). In der Klänge Liste erscheint der neue Klang:



- Wechsle ins Skripte-Register.



- Wähle eine zusätzliche Figur: *Bananas*



- Schreibe für die Bananen-Figur folgendes Skript:



Wenn diese Figur angeklickt wird
 gehe zu vorderster Ebene
 gehe zu x: -190 y: -120
 sende essen an alle

sende Nachricht1 an alle

Neue Nachricht
essen
✓ Nachricht1

Wähle *neue Nachricht...* und schreibe *essen*.

- Schreibe für das Äffchen folgendes Skript:

Wenn ich essen empfangen
 gleite in 1 Sek. zu x: -190 y: -100
 spiele Klang Chomp
 warte 1 Sekunden
 gleite in 1 Sek. zu x: -60 y: 80

- Teste das Skript, indem du auf die Banane klickst.

Gib deinem Haustier zu trinken

- Wähle eine weitere Figur: *Glass Water*.
- Ändere die Größe des Glases bei den Figuren-Infos auf 70%.
- Schreibe für das Wasserglas folgendes Skript:



Wenn diese Figur angeklickt wird
 gehe zu vorderster Ebene
 gehe zu x: -80 y: -120
 sende trinken an alle
 warte 1 Sekunden
 wechsle zu Kostüm glass water-b
 spiele Klang Water Drop
 warte 1 Sekunden
 wechsle zu Kostüm glass water-a

Erstelle eine neue Nachricht.

Wechselt zum leeren Glas

Wechselt zum vollen Glas

- Das Äffchen bekommt noch folgendes Skript:

Wenn ich trinken empfangen
 gleite in 1 Sek. zu x: -80 y: -100
 warte 1 Sekunden
 gleite in 1 Sek. zu x: -60 y: 80

- Teste die Skripts, indem du auf das Wasserglas klickst.

Sprechen mit dem Äffchen

Damit das Äffchen verschiedene Sätze sagen kann, musst du eine Variable erstellen.

- Wähle in der Block-Palette *Variablen*.
- Klicke auf *Neue Variable*.
- Nenne die neu erstellte Variable *Satzwahl* und klicke danach auf *OK*.



- Das neu auftauchende orange Feld auf der Bühne stört hier eher. Klicke deshalb auf das Häkchen vor der Variablen. Es verschwindet danach und auch das orange Feld auf der Bühne.



- Schreibe folgendes, weiteres Skript für das Äffchen:

Wenn diese Figur angeklickt wird
 setze Satzwahl auf Zufallszahl von 1 bis 3
 falls Satzwahl = 1, dann
 sage Ich liebe Bananen. für 2 Sekunden
 falls Satzwahl = 2, dann
 sage Das kitzelt. für 2 Sekunden
 falls Satzwahl = 3, dann
 sage Wollen wir spielen? für 2 Sekunden

Wählt zufällig einen der drei Sätze aus.

- Teste das Skript und beobachte, was das Äffchen sagt, wenn es berührt wird.



Zeit zu spielen

- Wähle eine weitere Figur: *Trampoline*
- Schreibe für das Trampolin folgendes Skript:

```

Wenn diese Figur angeklickt wird
  gehe zu x: 130 y: -120
  sende spielen an alle
  
```

- Das Äffchen bekommt dieses zusätzliche Skript:

```

Wenn ich spielen empfangen
  gehe zu vorderster Ebene
  gleite in 1 Sek. zu x: 120 y: -20
  wiederhole 4 mal
    ändere y um 20
    warte 0.3 Sekunden
    ändere y um -20
    warte 0.3 Sekunden
  gleite in 1 Sek. zu x: -60 y: 80
  
```

Eine positive Zahl lässt das Äffchen nach oben springen.

Eine negative Zahl lässt das Äffchen nach unten springen.

- Teste das Skript, indem du auf das Trampolin klickst.

Wie hungrig ist dein Äffchen?

Du kannst den Hunger deines Äffchens mit einer Zahl anzeigen lassen.

- Erstelle eine neue Variable und gib ihr den Namen *Hunger*. Lasse das Häkchen vor dieser Variable stehen, damit das Hungergefühl des Äffchens angezeigt wird.



- Schreibe folgende beiden Skripts für das Äffchen:

```

Wenn grüne Flagge angeklickt wird
  setze Hunger auf 0
  wiederhole fortlaufend
    ändere Hunger um 1
    warte 5 Sekunden
  
```

Stellt das Hungergefühl am Anfang auf 0.

Vergrößert das Hungergefühl alle 5 Sekunden um 1.

```

Wenn ich essen empfangen
  ändere Hunger um -1
  
```

Verringert das Hungergefühl nach jeder Mahlzeit um 1.

- Teste die Skripts: Beobachte, wie das Hungergefühl größer wird. Verkleinere es, indem du das Äffchen fütterst (auf die Bananen klickst).



Jetzt bist du dran!

Hast du eine gute Idee, wie du das jetzt Gelernte in einem Scratch-Projekt umsetzen könntest? Dann programmiere sie!

Ansonsten kannst du folgende Vorschläge ausprobieren:

- Lasse das Äffchen in diesem Projekt richtig sprechen, wenn es angeklickt wird, statt dass die Sätze nur eingeblendet werden (Erinnere dich an das zweite Projekt).
- Erstelle ein neues Projekt: Lass die Figur *Ballerina* auf dem Trampolin springen und sich in verschiedene Posen werfen. Auf Klick der Taste *d* soll sie das Rad schlagen (eine Umdrehung machen). Vielleicht schaffst du es noch, dass das Trampolin jeweils ein passendes Geräusch von sich gibt, wenn die *Ballerina* darauf springt.
- Erstelle ein neues Projekt: Kannst du eine Stoppuhr (Variable) so programmieren, dass sie nach dem Klick auf die grüne Flagge beginnt, die Sekunden zu zählen (nach dem Warten von einer Sekunde jeweils 1 zur Variable dazuzählen). Auf einen Klick der Taste *s* soll sie stoppen, auf einen Klick der Leerschlag-Taste soll die Anzeige auf 0 zurückgestellt werden.



Projekt starten

- Starte ein neues Projekt:

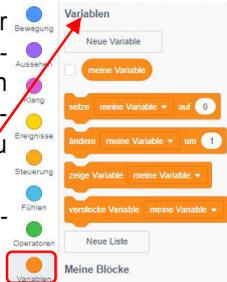


- Gib dem Projekt den Namen *Lass Blumen spriessen*.



Eine Steuerung bauen

In diesem Projekt soll der Benutzer zwischen verschiedene Blumenvarianten auswählen können. Um ein passendes Auswahl-Skript programmieren zu können, benötigst du als Erstes eine passende Variable.



- Wähle in der Block-Palette *Variablen*.
- Klicke auf *Neue Variable*.
- Nenne die neu erstellte Variable im auftauchenden Feld *Blumenvariante* und klicke danach auf *OK*.



- Das auf der Bühne neu auftauchende orange Feld passt in der linken oberen Ecke recht gut. Doch wäre es möglich, es auf der ganzen Bühne zu verschieben.

- Schreibe folgendes Skript für die **Bühne**:



Die Schleife stellt sicher, dass nicht nur bei Programmstart, sondern während des ganzen Spiels überprüft wird, ob eine Taste gedrückt wird.

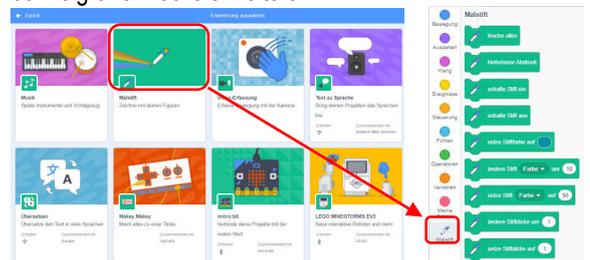
- Teste das Skript indem du auf die Zahl 1, 2 oder drei drückst, nachdem du auf die grüne Flagge geklickt hast. Die orange Anzeige auf der Bühne sollte sich daraufhin entsprechend ändern.



Erweiterung aktivieren

Um ein Blumenmeer auf die Bühne zaubern zu können, benötigst du noch ein paar zusätzliche Blocks. Diese findest du bei den *Erweiterungen*.

- Klicke auf das Erweiterungen-Feld.
- Bei den gezeigten Erweiterungen wählst du durch Klicken *Malstift* aus. Daraufhin wird die Block-Sammlung um neun dunkelgrüne Blocks erweitert.



Die Figur zeichnen und zeichnen lassen

Für dieses Projekt nimmst du keine vorgegebenen Figuren, sondern zeichnest sie selber:

- Lösche die Katzen-Figur mit Rechtsklick und wähle stattdessen eine leere Figur. Nenne sie 1.



- Stelle bei Füllfarbe mit den Schiebereglern rot ein und bei der Randfarbe nichts.



- Wähle das Kreiswerkzeug und zeichne einen etwa sechs Häuschen grossen Kreis. Beachte: Wenn du während des Zeichnens die Shift-Taste drückst, wird aus dem Kreis keine Ellipse. Um genau arbeiten zu können, kannst du die Arbeitsfläche vergrößern:



- Ziehe anschliessend mit dem *Auswählen*-Werkzeug den farbigen Kreis genau auf den Mittelpunkt der Zeichnungsfläche.



- Wechsle ins *Skripte*-Register.
- Schreibe das folgende Skript. Es bewirkt dass die Figur die Blume selber zeichnet, wenn du irgendwo auf die Bühne klickst.

Hier *Neue Nachricht* wählen. Danach *Blumenvariante 1* eingeben.

Legt die Breite der Blütenblätter fest.

Bestimmt die Anzahl Blütenblätter.

Legt die Länge der Blütenblätter fest.

Ändert die Farbe der Blütenmitte.

Ändert die Farbe der Blütenblätter.

- Wenn du jetzt dein Skript testest, wirst du immer noch einen weissen Bildschirm erhalten.
- Schreibe deshalb bei der Bühne noch das fehlende Skript:

- Teste die Skripts (Starte das Projekt. Drücke auf die 1, damit diese im orangen Variablen-Feld erscheint. Klicke danach irgendwo auf die Bühne.)

Sauberen Tisch machen

Irgendwann wird es zu viel mit den Blumen. Es muss deshalb noch ein Skript her, welches die Bühne wieder frei fegt.

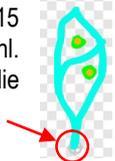
- Gehe deshalb zur **Bühne** und schreibe folgendes Mini-Skript:

- Teste die Leertaste!

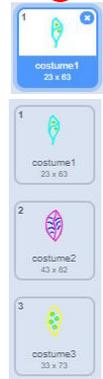
Eine zweite Blumenvariante

- Mit einem zweiten Skript kannst du ein weiteres Blumenmuster kreieren.
- Wähle eine zweite, leere Figur. Nenne sie 2.

- Wähle das Pinselwerkzeug aus und stelle dessen Dicke auf 6 sein.
- Zeichne eine ähnlich grosse Figur (etwa 15 Häuschen gross) mit den Farben deiner Wahl. Achte darauf, dass das untere Ende auf die Bühnenmitte zu liegen kommt.



- Auf der linken Seite siehst du nun dein gezeichnetes Kostüm.
- Erstelle nun ein zweites und ein drittes leeres Kostüm, indem du unten **links** auf den *Kostüm wählen*-Button klickst.
- Male auch auf diesen beiden Kostümen zwei Blütenblätter nach deinem Geschmack, aber in etwa der gleichen Grösse (15 Häuschen hoch). Auch sie sollen jeweils in der Bühnenmitte enden.
- Wechsle danach ins *Skripte*-Register.
- Schreibe das folgende Skript, welches aus deinen Blütenblättern selbständig Blumen zeichnet.



Stempelt 15mal dein Blütenblatt zu einer Blüte.

Legt die Anzahl Blütenblätter fest.

- Teste das Skript (Starte das Projekt. Drücke auf die 2, damit diese im orangen Variablen-Feld erscheint. Klicke danach irgendwo auf die Bühne.)



Aller guten Dinge sind drei

Runde dein Projekt mit einer dritten Blumensorte ab.

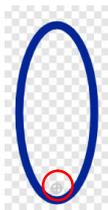
- Wähle eine dritte, leere Figur. Nenne sie 3.



- Wähle das Kreiswerkzeug

- Nimm folgende Einstellungen vor:

- Zeichne ein etwa 25 Häuschen hohes, blaues Oval. Die untere Rundung sollte dabei etwas über den Mittelpunkt der Zeichnungsfläche herausragen.



- Wähle den Fülleimer aus und stelle einen Farbverlauf ein:

- Fülle das Oval mit dem Farbverlauf.



- Wechsle danach ins *Skripte*-Register.
- Schreibe das Skript für die dritte Blume:

```

Wenn ich Blumenvariante3 empfangen
  gehe zu Mauszeiger
  setze Größe auf Zufallszahl von 50 bis 150
  wiederhole Größe / 5 mal
    hinterlasse Abdruck
    drehe dich um 150 Grad
    ändere Größe um -3.6
    ändere Effekt color um 0.8
  
```

- Teste das Skript (Starte das Projekt. Drücke auf die 3, damit diese im orangenen Variablen-Feld erscheint. Klicke danach irgendwo auf die Bühne.) Probiere auch andere Zahlen in den Blocks aus. Ändere sie, bis du deine optimale Blume geschaffen hast.

Saubere Bühne beim Start

Wenn die grüne Flagge geklickt wird, soll mit einer leeren Bühne gestartet werden können.

- Ergänze das Bühnenskript deshalb noch wie folgt:

```

Wenn grüne Flagge angeklickt wird
  lösche alles
  wiederhole fortlaufend
  
```

Die Blüten zur Geltung bringen

Auf weißem Hintergrund wirken die Blüten etwas fahl. Aber dem kann abgeholfen werden.

- Wechsle in das Register *Bühnenbilder*.
- Ziehe mit dem Rechteckwerkzeug eine schwarze Fläche über den ganzen Bühnenbereich.

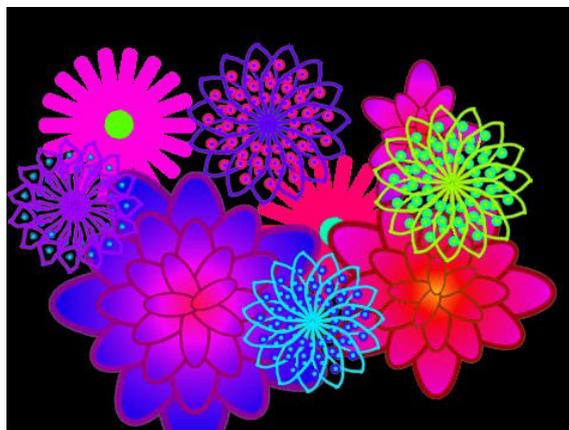


- Teste das Programm, indem du verschiedene Zahlen klickst und danach verschiedene Blumen zeichnest. Probiere auch die Leertaste aus!

Den Gesamteindruck abrunden

Da du in der Zwischenzeit sicher weißt, bei welcher Nummer welche Blüte erscheint, braucht es diese eher störende Anzeige nicht mehr.

- Wechsle ein letztes Mal ins *Skripte*-Register.
- Wähle bei den Blocks Variablen und klicke auf das Häkchen vor Blumenvariante. Mit ihm verschwindet auch die Anzeige auf der Bühne.
- Der letzte Test!



Jetzt bist du dran!

Hast du eine gute Idee, wie du das jetzt Gelernte in einem Scratch-Projekt umsetzen könntest? Dann programmiere sie!

Ansonsten kannst du folgende Vorschläge ausprobieren:

- Probiere bei deinem Projekt verschiedene Hintergründe aus. Versuche insbesondere ein paar Farbverläufe aus.
- Kannst du in diesem Projekt bei der zweiten und/oder dritten Blume noch mehr Kostüme einbauen?
- Kannst du in diesem Projekt bei der ersten Blume das Skript so abändern, dass jeweils abwechselungsweise ein kurzes und ein langes Blütenblatt gezeichnet werden, statt dass alle gleichlang werden?

Vorgestellte Projekte und verwendete Blocks													
	Gehen/pralle ab												
	Richtung/sich drehen zu												
	gleiten/Zeit												
	Koordinaten												
	Drehtyp												
	sagen/denken					x							
	zeigen/verstecken					x							
	Kostüm/Bühnenbild wechseln					x							
	Grafik-Effekte												
	Grösse ändern												
	nach vorne/hinten gehen					x							
	Klänge/Instrumente					x							
	Tempo/Lautstärke												
	Ereignis-Wenn (ohne «Start»)					x							
	Senden/empfangen												
	warten					x							
	wiederholen					x							
	Falls dann/falls dann - sonst					x							
	stoppen												
	Klon												
	Wird berührt?					x							
	Frage/Antwort												
	Taste gedrückt												
	Entfernung/Mausposition												
	Lautstärke (Eingabe)												
	Video												
	Stoppuhr/Zeit												
	Benutzername												
	Grundoperationen												
	Zufallszahl					x							
	Vergleiche												
	und/oder/nicht												
	verbinde												
	Zeichen... von/Länge von												
	Mathematische Funktionen/Runden												
	neuer Block												
	Variable					x							
	Liste												
	Malspuren weg, Stift ein/aus, Abdruck												
	Stiftfarbe/-stärke/-dicke												
	Text zur Sprache												
7 Verstecken und suchen	x												
8 Lass es fliegen	x												
9 Fangspiel	x					x							
10 Pong	x	x											
11 Ein Haustier programmieren	x		x										
12 Lass Blumen blühen	x	x											

Wie benütze ich diese Broschüre?

Anhand der hier vorgestellten Projekte kannst du viele der Scratch-Bausteine und deren Möglichkeiten kennen lernen. Beachte aber: Wenn du nur die Blocks zusammenschiebst, um möglichst schnell ein fertiges Projekt zu erhalten, wirst du von diesem Lehrgang wenig profitieren. Lies deshalb alle dazu gehörenden Erklärungen sorgfältig durch und mache dir immer wieder Gedanken, wie du das Gelernte in eigenen Projekten anwenden könntest und probiere deine Ideen aus! Solltest du keine Ideen haben, versuche, die vorgeschlagenen Fortsetzungsaufgaben zu lösen. So kannst du die notwendigen Erfahrungen sammeln, um ein selbständiger Scratcher zu werden.

Dieser Lehrgang basiert auf den Scratch Coding Cards für Scratch 2.0 (nicht mehr verfügbar): <https://scratch.mit.edu/uf/10/8/19>

