

Mehrere Animationen

Wenn ein Sprite im Spiel mehrere Animationen hat, kann man durch eine **Variable** steuern, welche Animation gerade ablaufen soll:

- Das **Bewegungsskript** setzt die Variable, z.B. auf „Laufen“.
- Das **Animationsskript** wählt entsprechend der Variable die richtigen Kostüme.

Im Beispiel soll die Katze drei verschiedene Animationszustände haben:

- **Laufen**, d.h. sie **wechselt** zwischen den Kostümen Nr. 1 bis 16. Die Katze läuft, während der Spieler die Tasten „links“ oder „rechts“ drückt.
- **Stehen**, d.h. sie zeigt Kostüm Nr. 17, und **gelegentlich** Kostüm Nr. 18 (so sieht es aus als würde sie ab und zu blinzeln). Die Katze steht, wenn der Spieler keine Taste drückt.
- **Sitzen**, d.h. sie zeigt die Kostüme Nr. 25 bis 27 und **bleibt** bei 27. Die Katze sitzt, solange der Spieler die Taste „unten“ drückt.

Betrachten zunächst nur die Zustände „Laufen“ und „Stehen“.

Wir verwenden eine Variable „Animation“. Das **Bewegungsskript** setzt diese entsprechend der Tasten, die der Spieler drückt.

- Da die Katze immer steht, wenn keine Taste gedrückt ist, wird die Variable zu Beginn der Wiederholschleife jedesmal auf „Stehen“ gesetzt.
- Wenn die Pfeiltaste links oder rechts gedrückt ist, wird sie zu „Laufen“ geändert.

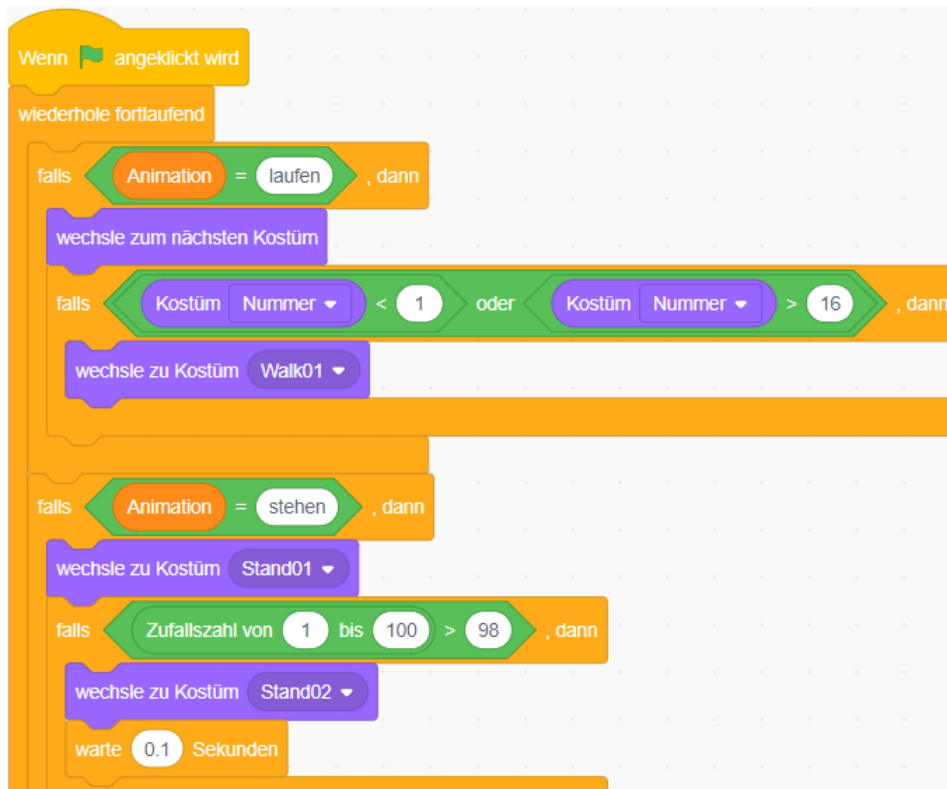
```

Wenn angeklickt wird
wiederhole fortlaufend
  setze Animation auf stehen
  falls Taste Pfeil nach links gedrückt? dann
    setze Animation auf laufen
    setze Richtung auf -90 Grad
    gehe 5 er Schritt
  falls Taste Pfeil nach rechts gedrückt? dann
    setze Animation auf laufen
    setze Richtung auf 90 Grad
    gehe 5 er Schritt
  
```



Das **Animationsskript** reagiert auf die Variable.

- Wenn die Variable den Wert „Laufen“ hat, wechselt das Skript durch die Kostüme Nr. 1 bis 16.
- Wenn die Variable den Wert „Stehen“ hat, wird das Kostüm 17 (Stand01) ausgewählt. Nur manchmal (in 2% der Fälle) wird stattdessen das Kostüm 18 (Stand02) angezeigt, und nach 0,1 Sekunde wieder Kostüm 17 (so wird das „Blinzeln“ erzeugt).



Einmalige Animation

Die Animation für das Hinsetzen ist im Vergleich zu den anderen besonders, da sie die Kostüme 26 bis 27 einmal durchläuft, dann aber bei Kostüm 27 bleibt – solange, bis die Katze in einen anderen Zustand kommt, z.B. Stehen oder Laufen.

In diesem Fall kann das **Animationskript** die einzelnen Kostüme nacheinander anzeigen.

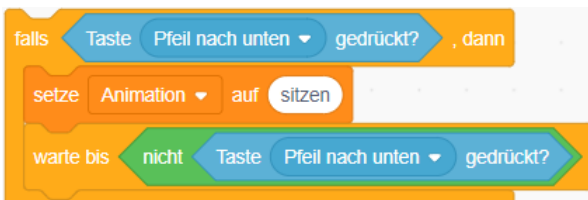
Um beim letzten Kostüm zu bleiben, ist der Block **warte bis ()** nützlich – das Skript wartet, bis der Animationszustand nicht mehr „Sitzen“ ist.



Auch für das **Bewegungsskript** kann man **warte bis ()** einsetzen:

Während die Katze sitzt, soll sie sich nicht weiter bewegen. Durch den Block **warte bis ()** wird die Wiederholschleife des Bewegungsskripts unterbrochen, solange, bis der Spieler die Taste „unten“ wieder loslässt. So reagiert das Skript nicht, wenn z.B. die Pfeiltaste links gedrückt wird.

Sobald der Spieler die Taste „unten“ loslässt, geht die Wiederholschleife weiter. Die Variable wechselt automatisch zum Wert „Stehen“ (oder „Laufen“, wenn der Spieler eine Pfeiltaste drückt). Die Bewegung kann weitergehen, und das Animationskript wählt ein anderes Kostüm.



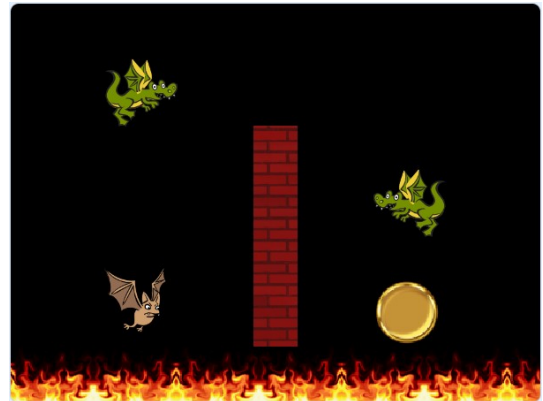
Aufgabe 1: Katze

Verwende die bereitgestellte Scratch-Vorlage, die alle Kostüme für die Figur der Katze enthält. Programmiere die Katze entsprechend der obigen Erläuterungen:

- Laufen bei Pfeiltaste links / rechts
- Hinsetzen bei Pfeiltaste unten
- Hochspringen bei Pfeiltaste oben (bewege die Katze 10 mal 10 Schritte nach oben, danach 10 mal 10 Schritte nach unten). Es gibt je ein Kostüm für das Springen bzw. das Fallen.
- Stehen, wenn keine Taste gedrückt ist, und dabei ab und zu blinzeln.

Aufgabe 2: Fledermaus

Verwende die bereitgestellte Vorlage. Sie enthält die Fledermaus mit Kostümen, in denen sie ihre Flügel bewegt, und weitere, in denen sie sich in Rauch auflöst, sowie weitere Figuren.

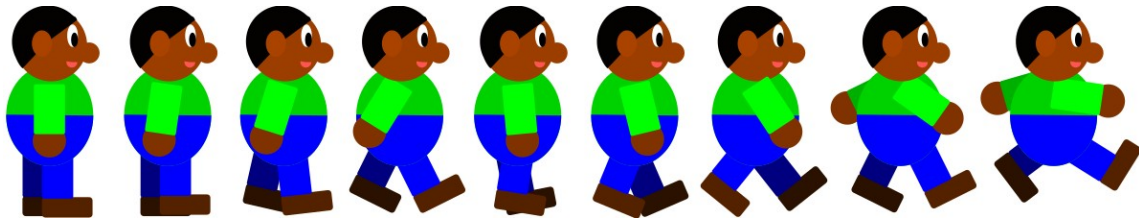


- Programmiere das **Flugverhalten** der Fledermaus mit der zugehörigen Animation:
 - Wenn keine Taste gedrückt ist, sinkt sie langsam abwärts. Dabei zeigt sie nur das Kostüm mit den Flügeln nach oben.
 - Bei Druck der Pfeiltaste nach oben fliegt sie langsam aufwärts. Dabei flattert sie mit den Flügeln.
 - Bei den Pfeiltasten links / rechts bewegt sie sich nach links / rechts. Dabei sinkt bzw. steigt sie weiterhin, je nachdem ob die Pfeiltaste nach oben gedrückt ist. Die Animation wird davon nicht beeinflusst (hängt nur vom Sinken bzw. Fliegen ab).
- Füge die beiden Drachen als **Gegner** hinzu. Sie fliegen immer von links nach rechts. Wenn die Fledermaus diese berührt, löst sie sich in Rauch auf (dabei sollte sie sich nicht mehr bewegen). Das letzte Kostüm (nach dem Auflösen) ist komplett durchsichtig, d.h. du musst sie nach der Animation nicht verstecken. Wenn du möchtest, kannst du auch eine kleine Animation für die Drachen hinzufügen: Es gibt ein Kostüm, in dem sie Feuer speien, das z.B. zufällig ab und zu gezeigt werden kann.
- Falls du Zeit und Lust hast, kannst du noch die **Spielmechanik** programmieren: Die Fledermaus muss vom Startplatz um das Hindernis herum fliegen bis zur Goldmünze. Sie darf weder das Hindernis noch das Feuer am Boden berühren (dort löst sie sich in Rauch auf).

Spritesheets

Bei der Spieleentwicklung beansprucht nicht nur die Programmierung, sondern auch die Entwicklung der Grafik viel Zeit. Die Arbeit wird meist von Grafikern übernommen, während Programmierer den Programmcode entwickeln. Gute Sprites zu zeichnen ist aufwendig und erfordert einiges Geschick, denn Animationen aus Einzelbildern sollen ja realistisch aussehen.

Um das Programmieren zu lernen, kann man sich aber glücklicherweise im Internet bedienen: Sogenannte Spritesheets enthalten Einzelbilder von Figuren, oft aus populären Computerspielen.



Um ein Spritesheet in Scratch zu verwenden, musst du die Einzelbilder (z.B. mit Paint.NET) ausschneiden und nacheinander als Kostüme einer Figur in dein Scratch-Projekt hochladen. Das bedeutet einige Arbeit – im Vergleich dazu, die Kostüme selbst zu zeichnen, ist das jedoch wenig.

Beachte, dass die viele Spritesheets **Copyright-geschütztes** Material enthalten. Solltest du dein Projekt verkaufen wollen, müsstest du erst die Einverständniserklärung der Besitzer einholen, also in der Regel vom Hersteller des Spiels – oder eben doch selbst zeichnen. Aber zum Üben ist es völlig in Ordnung, im Web z.B. nach „spritesheet super mario“ oder ähnlichem zu suchen und die Grafiken in dein Spiel einzubauen.