

Sternsammler programmieren

Programmiere dein erstes Spiel in Scratch: Sternsammler ist eine rasante Schatzsuche unter Wasser. Befolge einfach die Schritte in diesem Kapitel, dann kannst du einen Freund zum Spiel auffordern. Schafft er es, dich zu schlagen?

ZIEL DES SPIELS

Ziel des Spiels ist es, möglichst viele goldene Sterne zu sammeln, und das erledigst du mit der Katze. Pass auf die tödlichen Tintenfische auf! Du musst dich sehr schnell bewegen. Die wichtigsten Figuren im Spiel sind unten aufgeführt.



◁ Die Katze

Du bewegst sie mit der Maus, denn die Figur der Katze folgt einfach dem Mauszeiger.



◁ Oktopusse

Die Oktopusse streifen durch das Meer, aber sie sind langsamer als du. Berührt dich ein Tintenfisch, ist das Spiel aus!



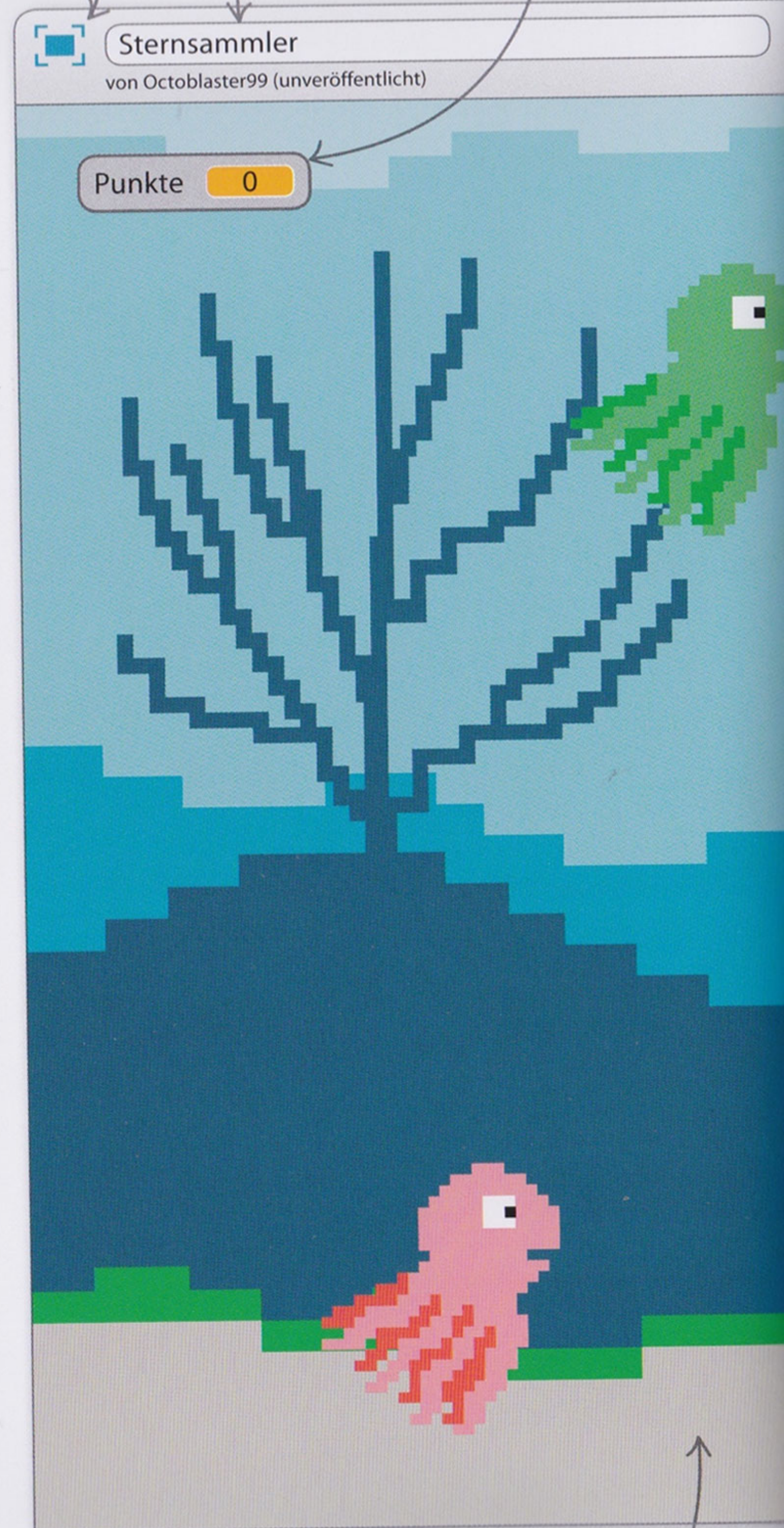
◁ Sterne

Sie tauchen einzeln an willkürlichen Orten auf. Berührst du einen, erhältst du einen Punkt.

Klicke hier, dann füllt die Bühne den Bildschirm aus.

Feld für den Programmnamen

Die Punktzahl zeigt, wie viele Sterne du gesammelt hast.



Das Unterwasser-Bühnenbild sorgt für die richtige Atmosphäre.

Punkte erhältst du, indem du Sterne sammelst.

Ein Klick auf die grüne Fahne startet ein neues Spiel.



Du bist im Spiel die Katze. Die Figur bewegt sich mit der Maus.

SPIELSTEUERUNG

Das Spiel wird mit der Maus oder mit dem Touchpad gesteuert.



Ein Klick auf das Stoppsymbol beendet das Spiel.

Berühre niemals die drei Tintenfische! Jeder bewegt sich auf seine eigene Art und Weise.

◁ Unter Wasser

Sternsammler hat einen Unterwasser-Hintergrund, du kannst aber auch ein anderes Bild auswählen, z. B. das Weltall oder ein Foto deines Zimmers.

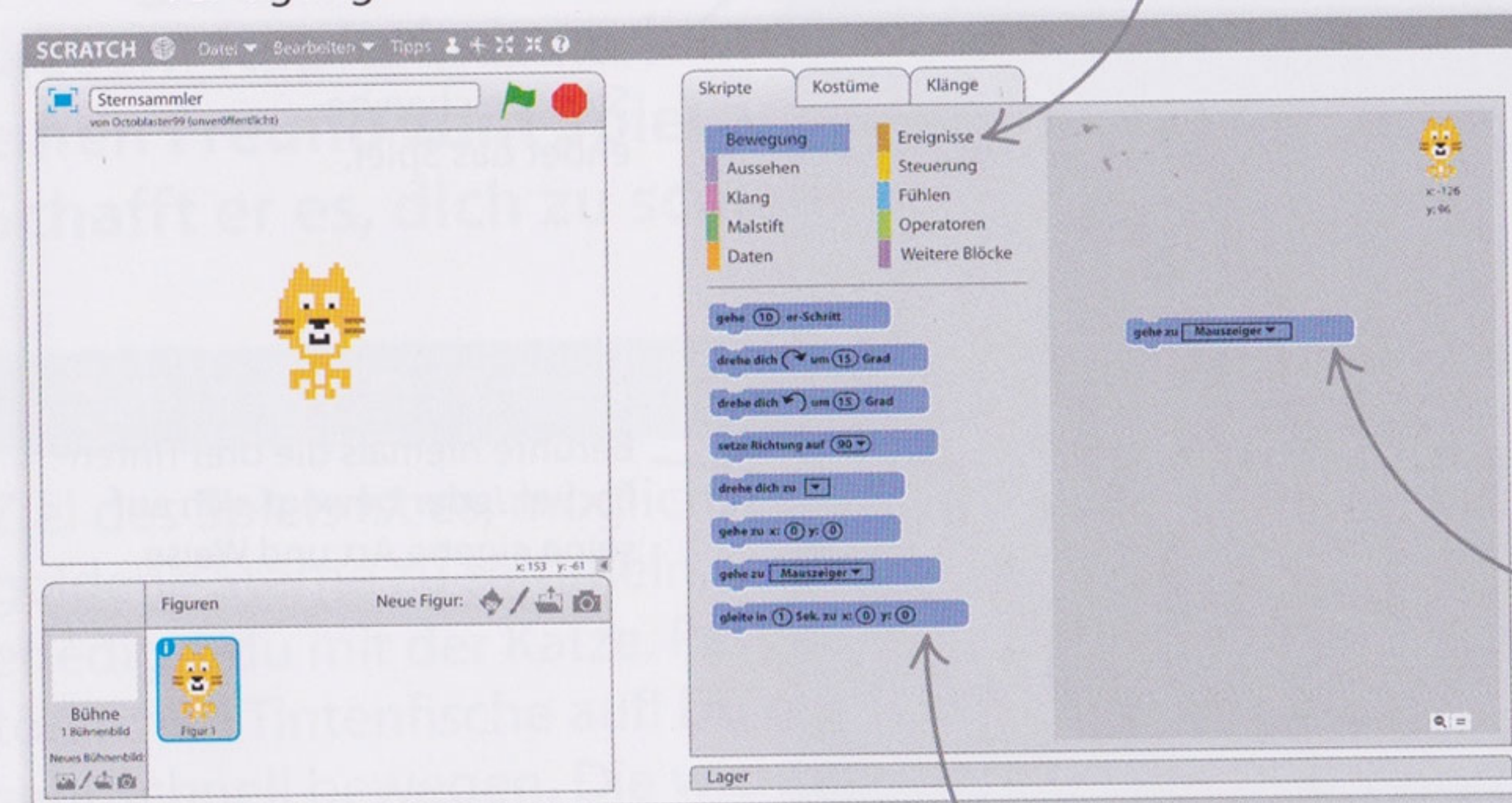
Fertig?
Los!



Skripte erstellen

Wie jedes Programm in Scratch wird auch Sternsammler aus einzelnen farbigen Blöcken zusammengestellt. Jeder Block ist eine Anweisung, die eine Figur ausführen muss. Wir beginnen mit der wichtigsten Figur im Spiel: der Katze.

- 1** Starte Scratch und klicke entweder auf „Entwickeln“ oder auf „Datei“ und „Neu“. Das Fenster sieht aus wie hier gezeigt. Die Katze ist bereit und in der Mitte stehen blaue Blöcke zur Verfügung.



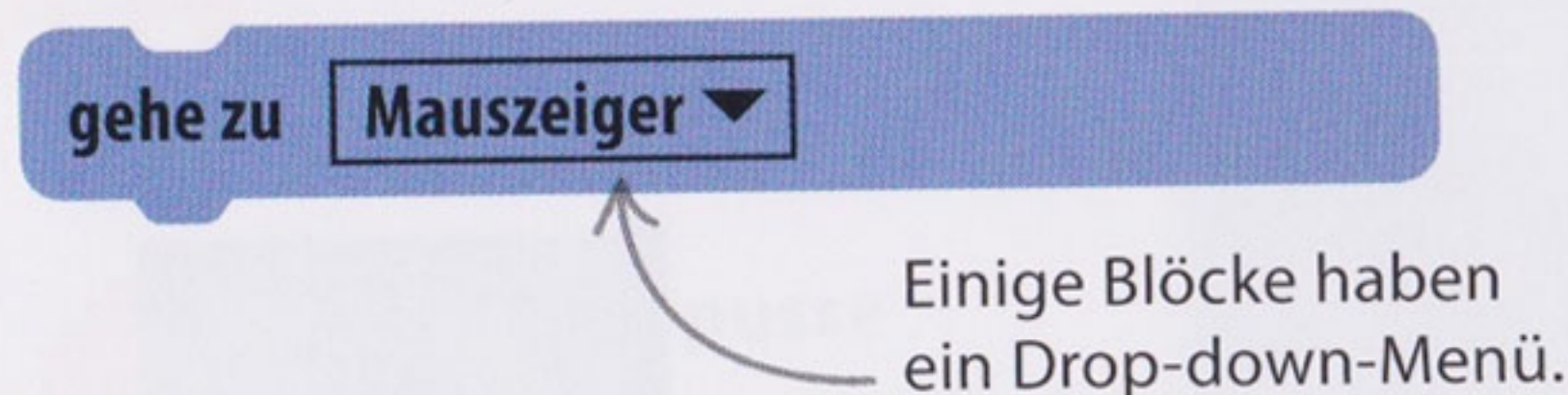
Wenn du hier klickst, siehst du die verschiedenen Arten von Blöcken.



Ziehe die benötigten Blöcke in diesen Bereich. So stellst du ein Skript zusammen.

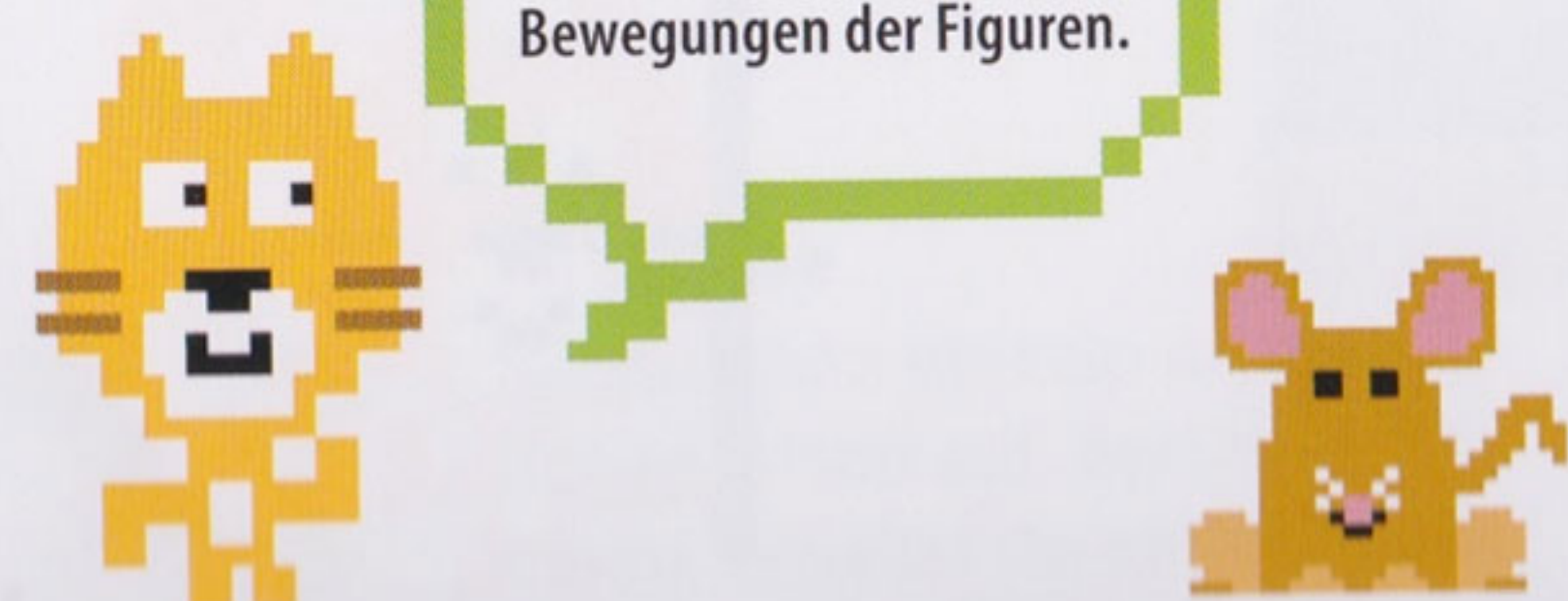
- 2** Wir programmieren die Katze so, dass sie sich dorthin bewegt, wo der Spieler die Maus hinführt. Klicke auf den Block „gehe zu Mauszeiger“ und ziehe ihn in den rechten Teil des Bildschirms – den Skriptbereich.

Wähle Blöcke aus der Liste in der Mitte aus.

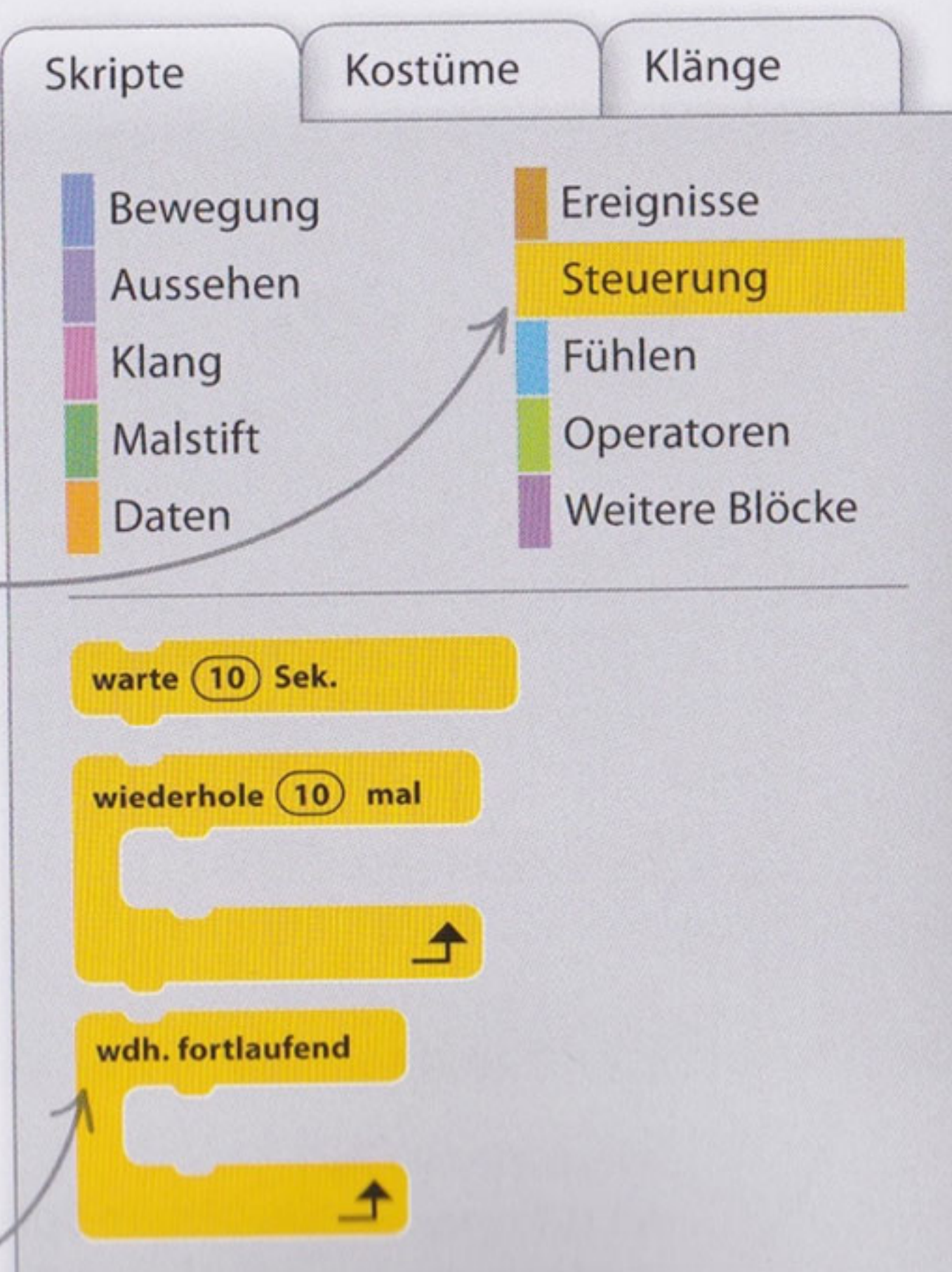


Einige Blöcke haben ein Drop-down-Menü.

Die blauen Bewegungs-Blöcke steuern die Bewegungen der Figuren.



- 3** Klicke auf „Steuerung“ und den Block „wiederhole fortlaufend“.



Wenn du auf Steuerung klickst, werden die gelben Blöcke angezeigt.

Ziehe den Block „wiederhole fortlaufend“ in den Skriptbereich.

- 4** Ziehe ihn nach rechts und setze ihn über dem blauen Block ab, sodass er ihn umschließt:



- 5** Klicke auf die braune Schaltfläche „Ereignisse“ und suche den Block mit der grünen Fahne. Ziehe ihn nach rechts und setze ihn an den Anfang des Skripts. Lies das Skript und überlege, was die Blöcke bewirken:



Der Block startet das Spiel nach dem Klick auf die grüne Fahne.

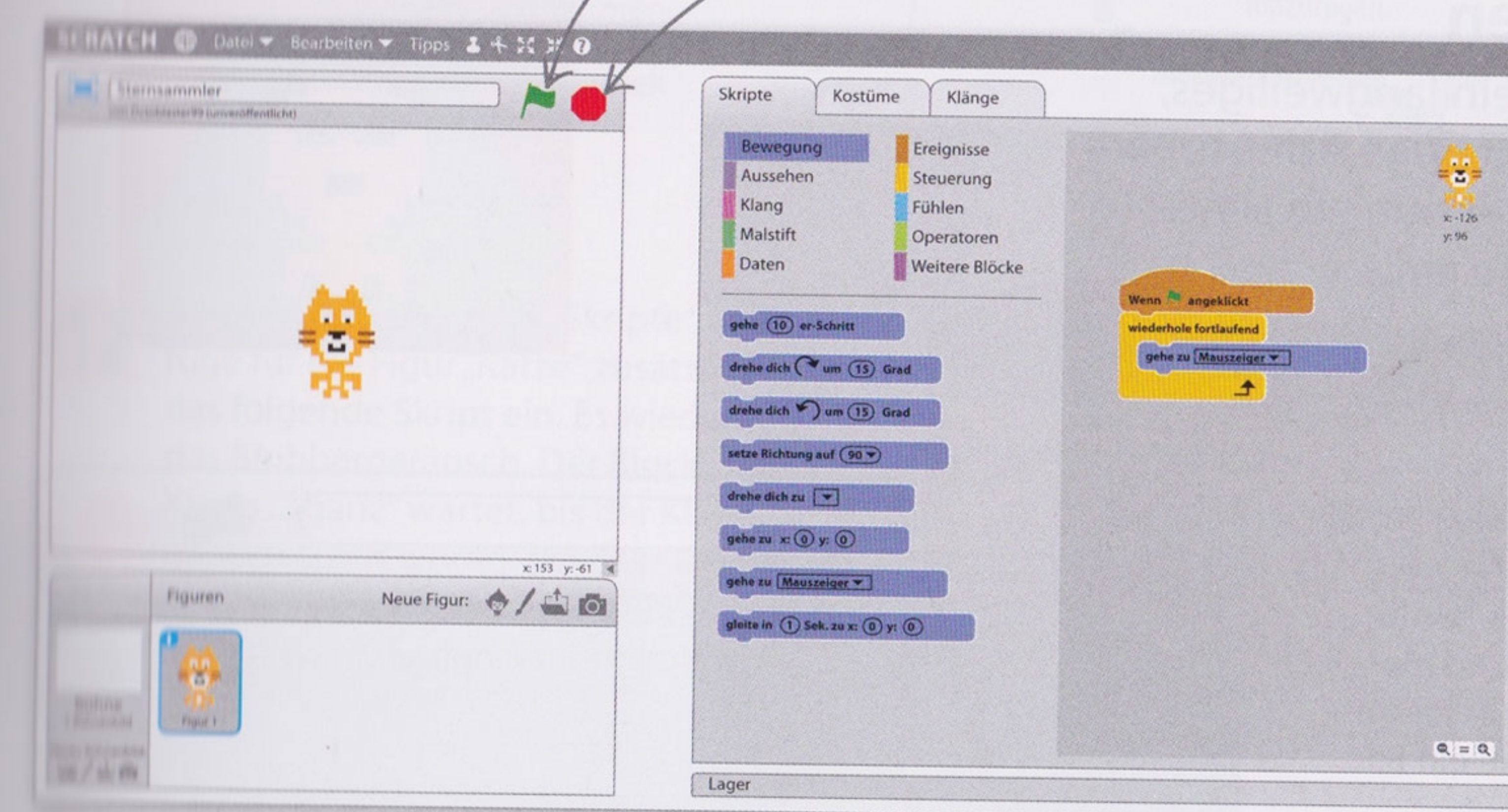
Dieser Block sorgt dafür, dass sich die Katze mit dem Mauszeiger bewegt.

Dieser Block bewirkt, dass der blaue Block laufend wiederholt wird.

- 6** Rechts oben über der Bühne siehst du die grüne Fahne. Wenn du sie anklickst, wird das Skript ausgeführt.

Ein Klick hierauf startet das Spiel.

Ein Klick auf die rote Stopp-Schaltfläche beendet das Skript.



- 7** Bewege die Maus und beobachte, was passiert: Die Katze bewegt sich zusammen mit dem Mauszeiger über die Bühne.



► **Gut gemacht!**
Du hast dein erstes Scratch-Skript erstellt. Nun fügen wir weitere Dinge hinzu, damit ein ganzes Spiel entsteht.



- 8** Die arme Katze heißt einfach „Figur1“. Das sollten wir ändern. Markiere „Figur1“ (die Katze) in der Liste der Figuren und klicke auf das blaue „i“ in der Ecke, um die Informationen aufzurufen. Gib „Katze“ in das Feld ein.

Wenn du hier klickst, werden die Informationen angezeigt.

Ein Klick auf den Pfeil schließt das Fenster wieder.

Gib hier den Namen der Figur ein.

Katze

x: 84 y: -69 Richtung: 90°

Drehmodus:

ziehbar im Präsentationsmodus: ☐

sichtbar: ☒

Der neue Name wird angezeigt.

Die Bühne gestalten

Im Moment ist die Bühne ein langweiliges, weißes Rechteck. Um die richtige Atmosphäre zu schaffen, fügen wir daher nun ein Hintergrundbild und Klangeffekte ein.



- 9** Links neben der Figurenliste ist eine Schaltfläche, mit der du Bilder aus der „Bühnenbild-Bibliothek“ einfügen kannst. Klicke darauf und markiere das Bild „underwater2“. Klicke auf „OK“. Nun hat die Bühne ein Bild.

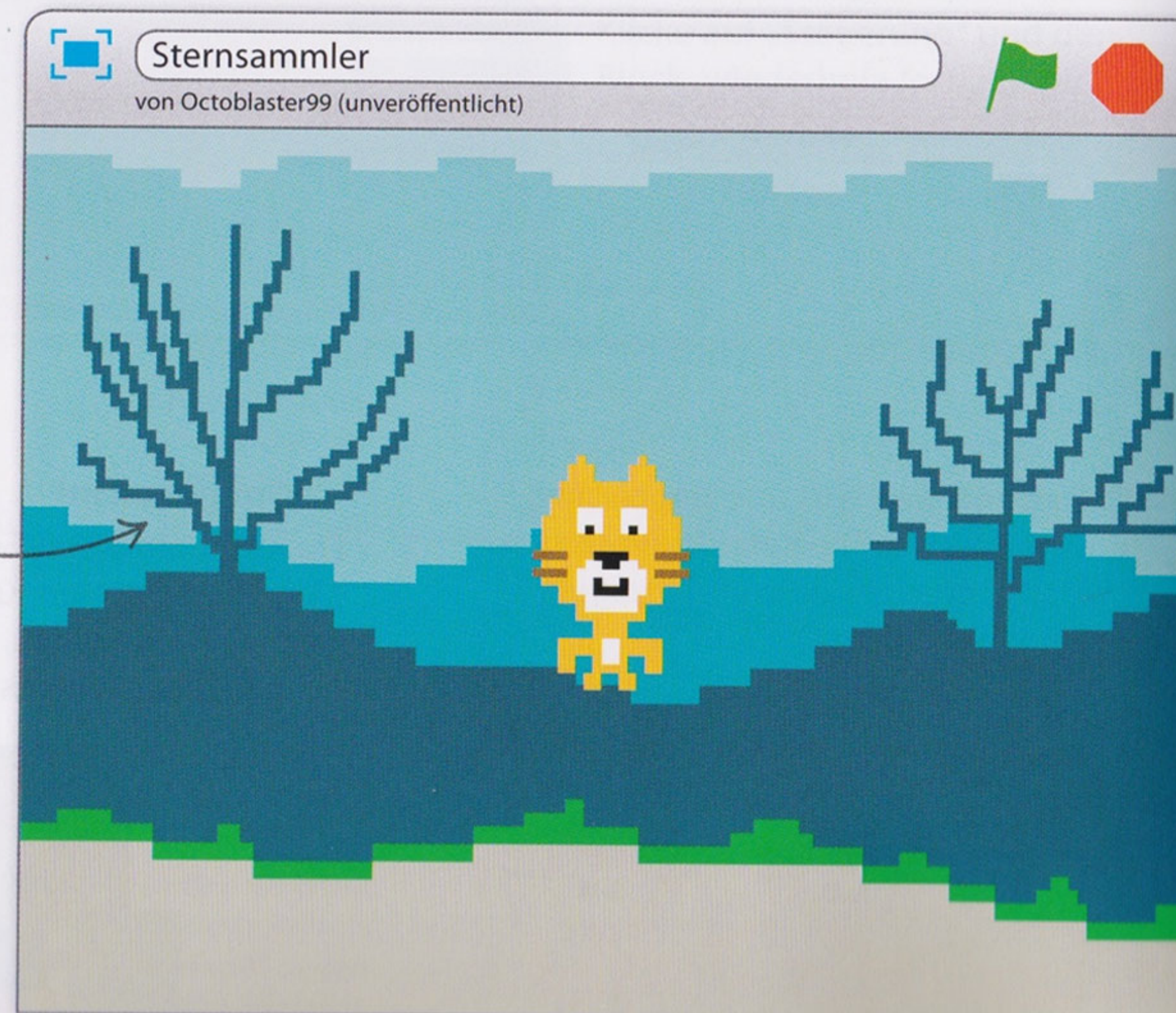
Das Bühnenbild ist reine Dekoration. Es hat keine Wirkung auf die Figuren.

Bühne

1 Bühnenbild

Neues Bühnenbild:

Ein Klick hier öffnet die Bühnenbild-Bibliothek.



Soundeffekte

Nun fügen wir für die Katze ein Blubbergeräusch ein, damit es so klingt, als bewege sie sich unter Wasser.

- 10** Markiere die Katze in der Figurenliste und klicke über der Blockpalette auf das Register „Klänge“. Ein Klick auf das Lautsprecher-Symbol öffnet die Klangbibliothek.

Skripte Kostüme Klänge

Neuer Klang:

Klang aufnehmen.

Klang aus der Bibliothek auswählen.

Klang vom Computer einfügen.

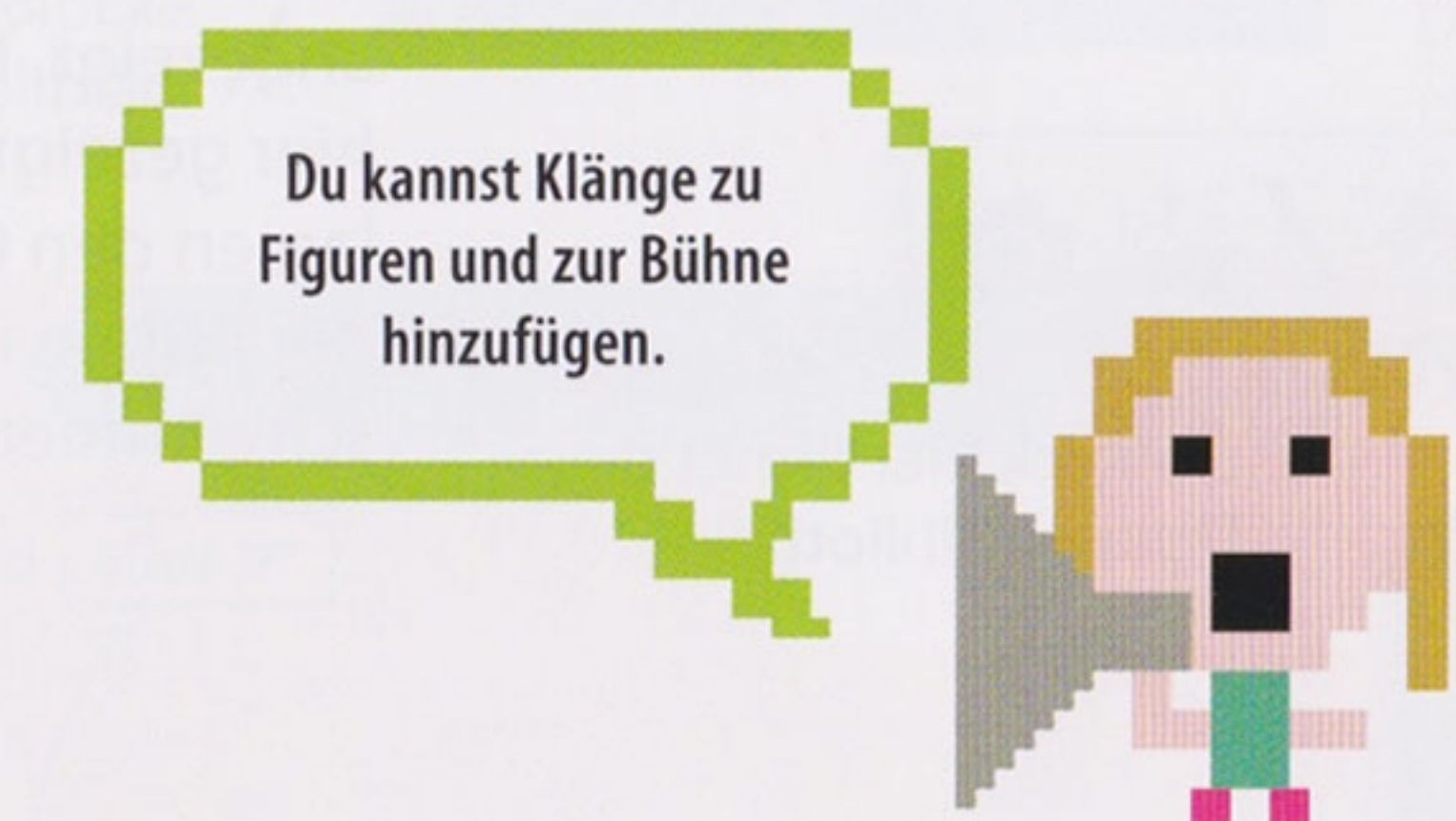
Register „Klänge“

- 11** Markiere in der Bibliothek den Klang „bubbles“. Klickst du auf „Play“, hörst du den Klang. Soll er ins Spiel eingefügt werden, klicke auf den Lautsprecher und auf „OK“. In der Klangliste wird nun „bubbles“ angezeigt.

Hier werden Klänge gelöscht.

bubbles 00:04.08

Die Dauer des Klangeffekts wird hier angezeigt.



- 12** Klicke auf das Register „Skripte“ und füge für die Figur „Katze“ zusätzlich das folgende Skript ein. Es wiederholt das Blubbergeräusch. Der Block „spiele Klang ... ganz“ wartet, bis der Klang ganz gespielt wurde, bevor er ihn neu startet. Wenn du das Spiel nun ausführst, hörst du den Soundeffekt.

Wenn angeklickt

wiederhole fortlaufend

spiele Klang bubbles ganz

Diesen Block findest du in der Blockpalette unter „Klang“.

EXPERTENTIPP

Schleifen

Eine Schleife ist ein Programmabschnitt, der mehrmals wiederholt wird. Der Block „wiederhole fortlaufend“ erzeugt eine Schleife, die endlos wiederholt wird. Es gibt aber auch andere Schleifen, bei denen du angeben kannst, wie oft eine Aktion wiederholt werden soll. Schleifen gibt es in fast allen Programmiersprachen.

Ausführung von oben nach unten

Wenn angeklickt

wiederhole fortlaufend

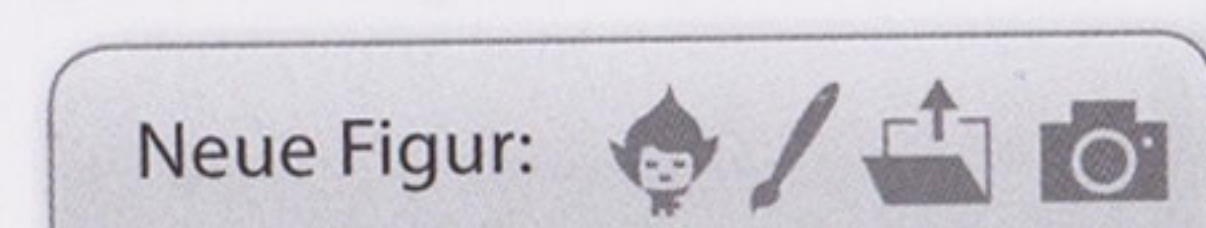
spiele Klang bubbles ganz

Der Block „wiederhole fortlaufend“ führt das Programm immer wieder zum Anfang des Blocks.

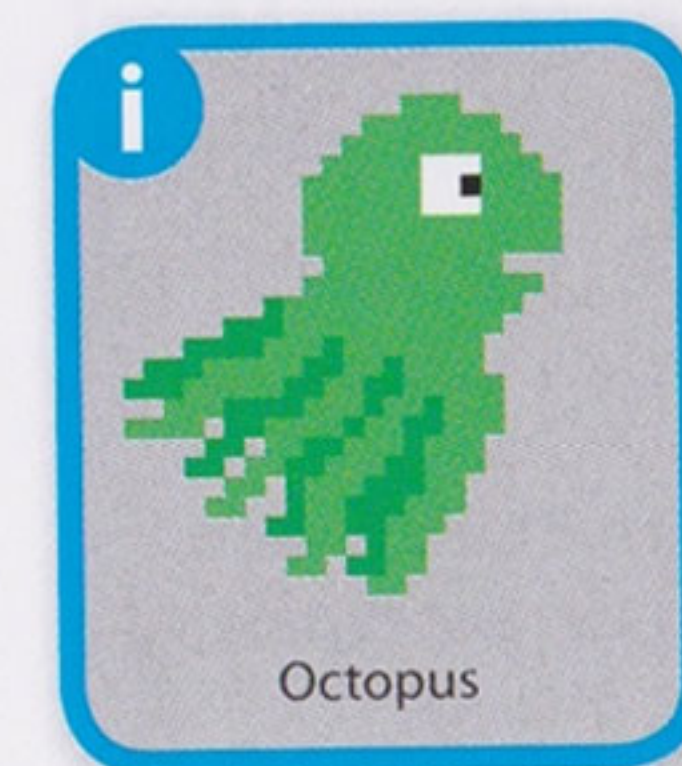
Ein Feind tritt auf

Damit das Spiel interessanter wird, brauchen wir einen Feind. Wie wäre es mit einem tödlichen Tintenfisch? Er bewegt sich auf der Bühne hin und her und der Spieler muss ihm ausweichen, sonst ist das Spiel vorbei.

- 13** Klicke auf die hier gezeigte Schaltfläche, um eine zweite Figur zu erstellen. Markiere in der Figuren-Bibliothek den Oktopus und klicke auf „OK“.



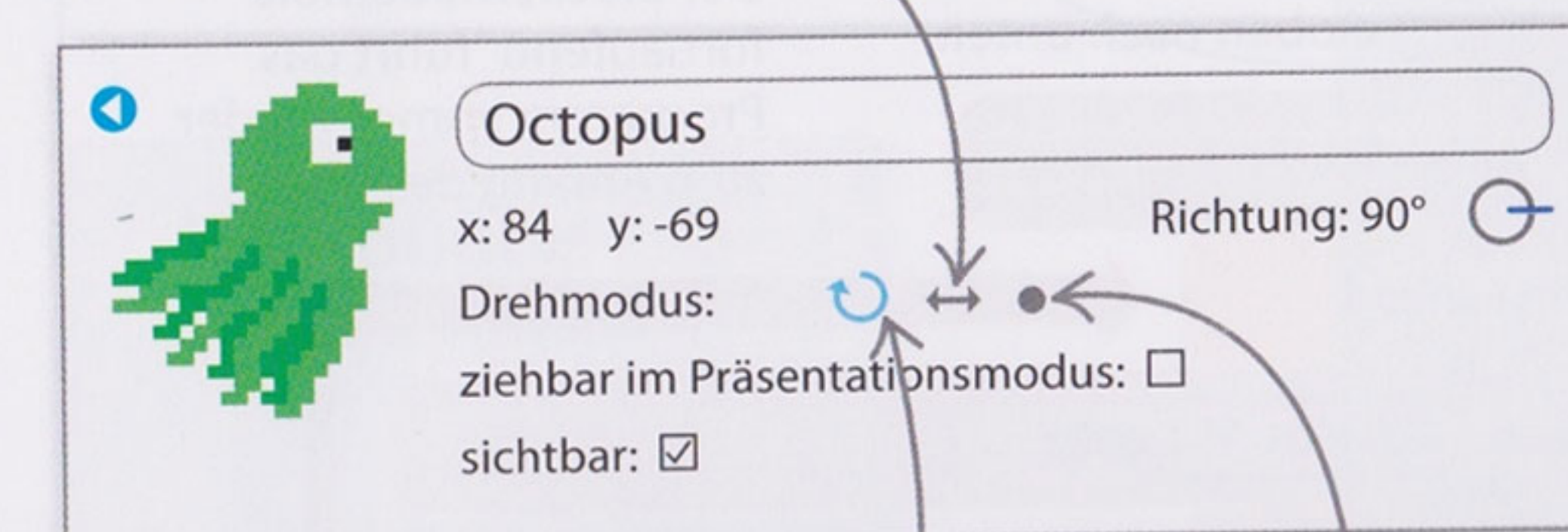
Ein Klick hier öffnet die Figuren-Bibliothek.



Die Oktopus-Figur erscheint in der Liste.

- 15** Führe das Skript aus. Der Tintenfisch bewegt sich hin und her, aber teilweise schwimmt er dabei mit dem Kopf nach unten. Das können wir ändern, indem wir ihm befehlen, anders zu wenden. Markiere den Oktopus und klicke auf das blaue „i“. In dem nun geöffneten Feld gibt es drei Optionen für den „Drehmodus“.

Bei Auswahl der mittleren Option dreht sich der Oktopus am Rand jeweils um.



Bei Wahl der linken Option stellt sich der Oktopus beim Abprallen auf den Kopf.

Bei Wahl der rechten Option prallt der Oktopus ab, ohne sich umzudrehen.

- 14** Gib der Oktopus-Figur das hier gezeigte Skript. Wenn du in der Blockpalette auf „Bewegung“ klickst, werden die blauen Blöcke angezeigt. Die beiden hier gezeigten Blöcke lassen den Oktopus auf der Bühne hin und her schwimmen.

Dieser Block führt das Skript aus, wenn das Spiel startet.

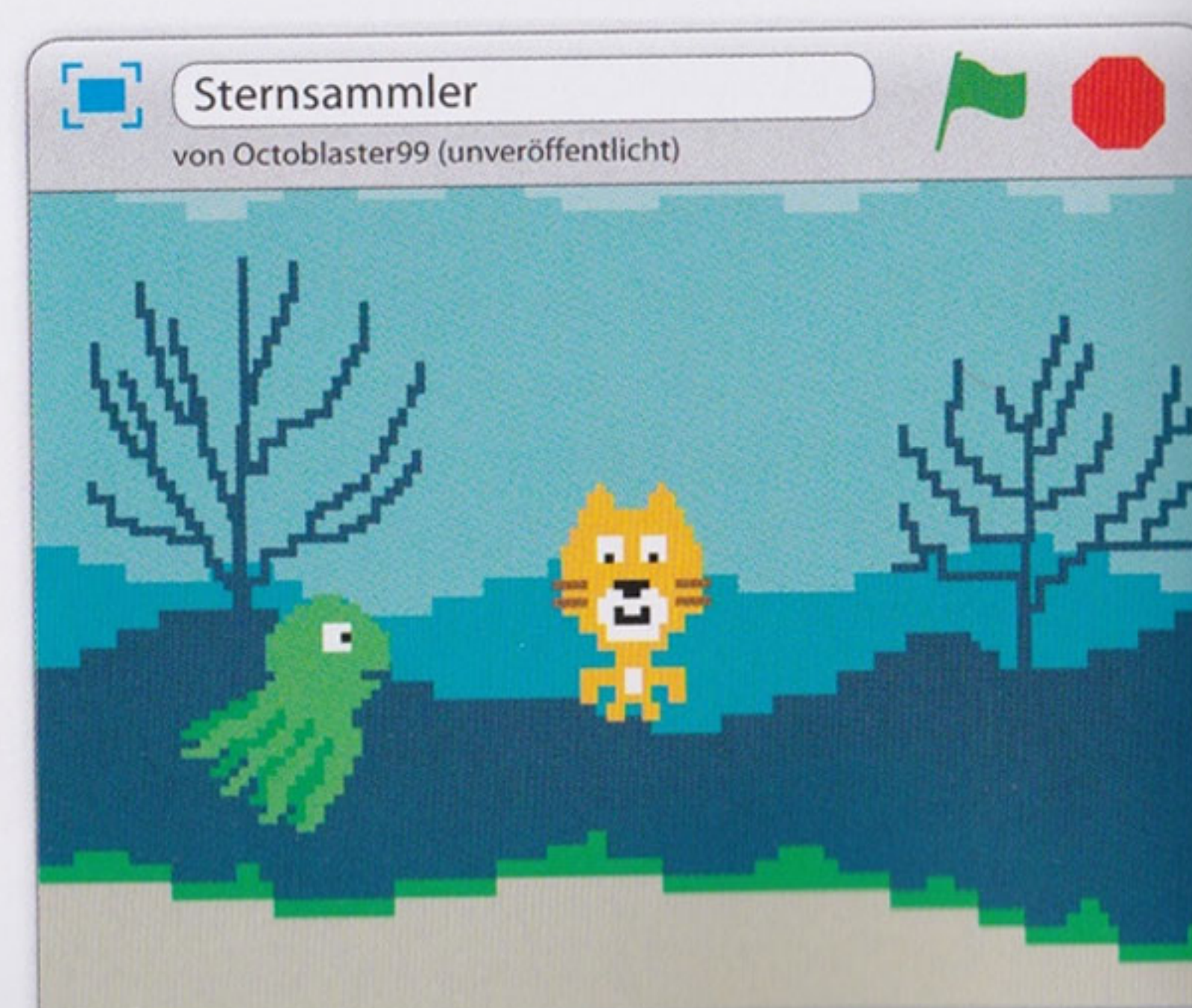
Die dunkelblauen Bewegungsblöcke steuern die Bewegungen der Figuren.



Der Block „wiederhole fortlaufend“ wiederholt die Aktionen ständig.

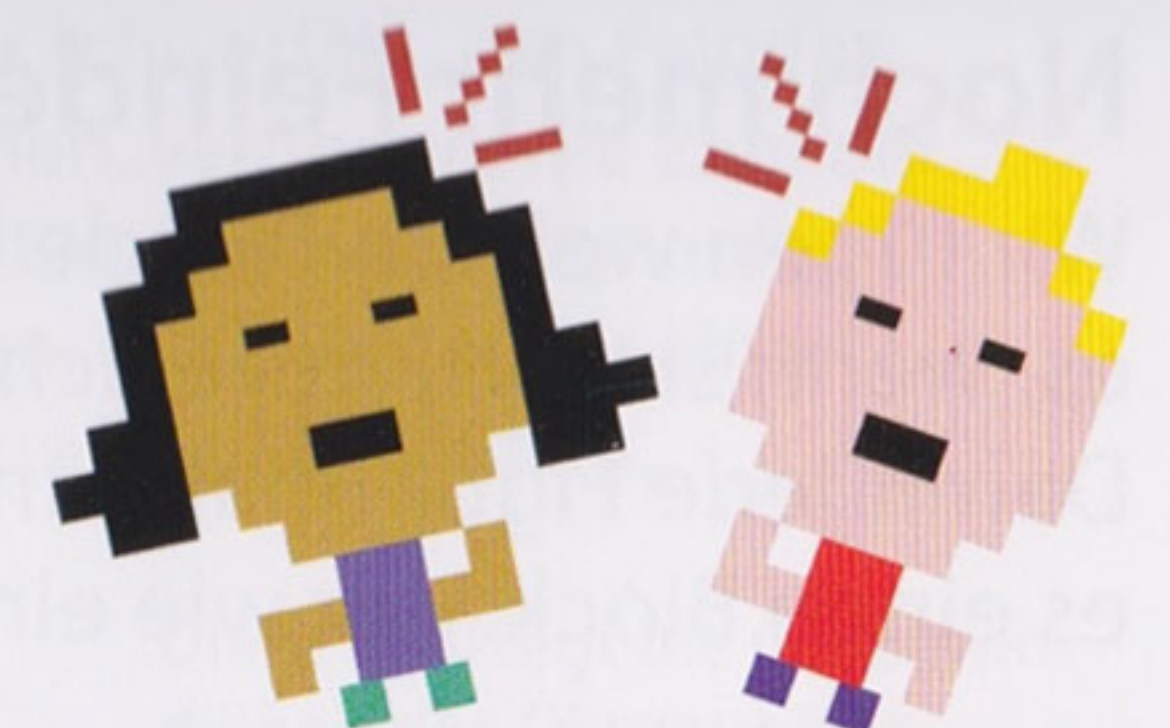
Dieser Block sorgt dafür, dass der Oktopus die Bühne nicht verlässt.

- 16** Wähle die mittlere Option und führe das Programm aus. Der Oktopus sollte nun aufrecht bleiben und immer nach vorne blicken. Du kannst ihn mit der Maus an die gewünschte Startposition ziehen.

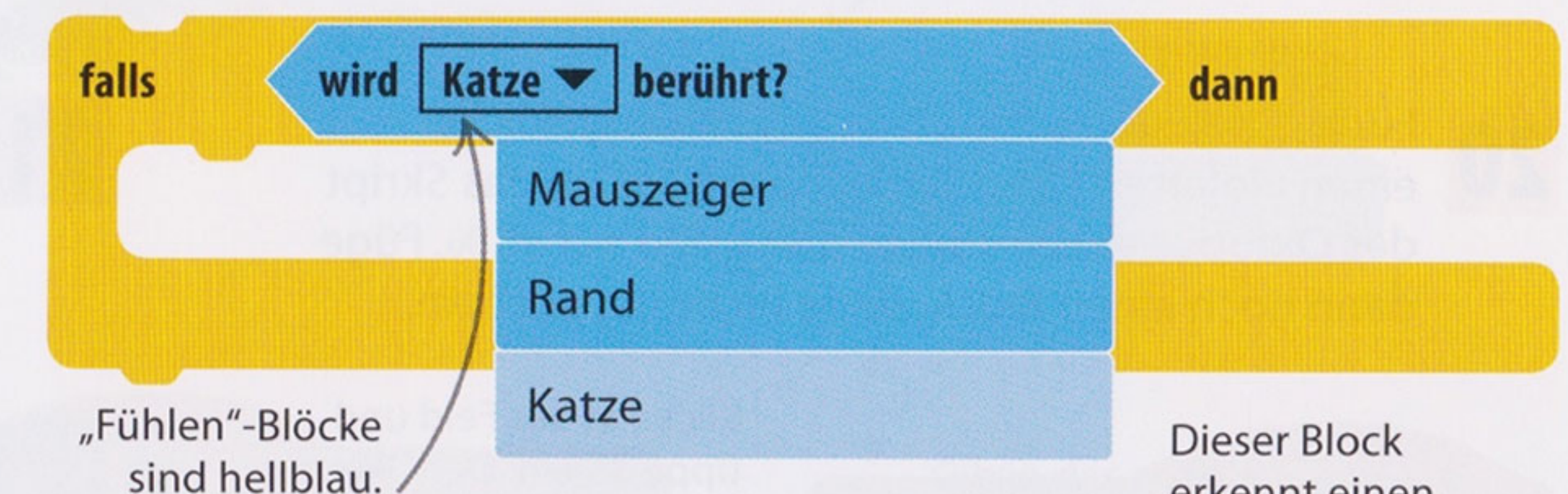


Zusammenstöße

Bisher bewegen sich Oktopus und Katze einfach „durcheinander“, ohne dass etwas geschieht. Mit einem neuen Skript bleiben sie beim Zusammenprall stehen. Die sogenannte Kollisionserkennung ist in Computerspielen sehr wichtig.



- 17** Markiere den Oktopus und ziehe einen gelben „falls-dann“-Block in den Skriptbereich. Füge nun einen hellblauen „Fühlen“-Block in den oberen Teil des „falls-dann“-Blocks ein. Klicke auf das Drop-down-Menü und dann auf „Katze“. Mit diesem Skript „fühlt“ der Oktopus die Katze.



„Fühlen“-Blöcke sind hellblau.

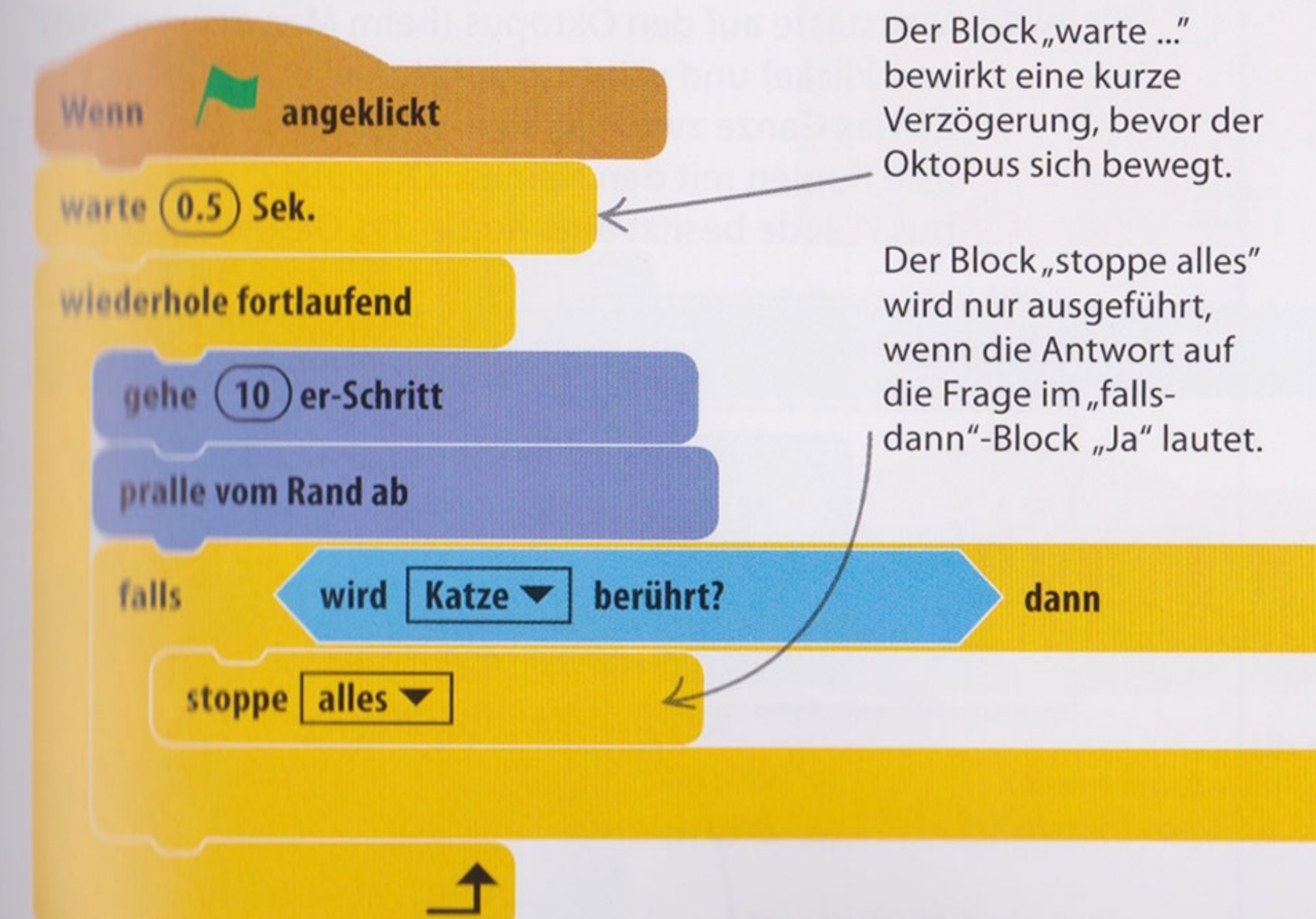
Dieser Block erkennt einen Zusammenstoß.

- 18** Klicke in der Blockpalette erneut auf „Steuerung“ und setze einen Block „stoppe alles“ in den „falls-dann“-Block ein. Er beendet alle Aktionen, sobald der Oktopus die Katze berührt, und das Spiel ist zu Ende.



Beendet das Spiel, wenn die Figuren kollidieren.

- 19** Füge nun den „falls-dann“-Block in das Hauptskript des Oktopus ein, und zwar direkt nach den blauen Bewegungsblöcken. Füge zudem vor der Schleife „warte 0.5 Sek.“ ein und prüfe, was passiert.



Der Block „warte ...“ bewirkt eine kurze Verzögerung, bevor der Oktopus sich bewegt.

Der Block „stoppe alles“ wird nur ausgeführt, wenn die Antwort auf die Frage im „falls-dann“-Block „Ja“ lautet.

EXPERTENTIPP „falls-dann“

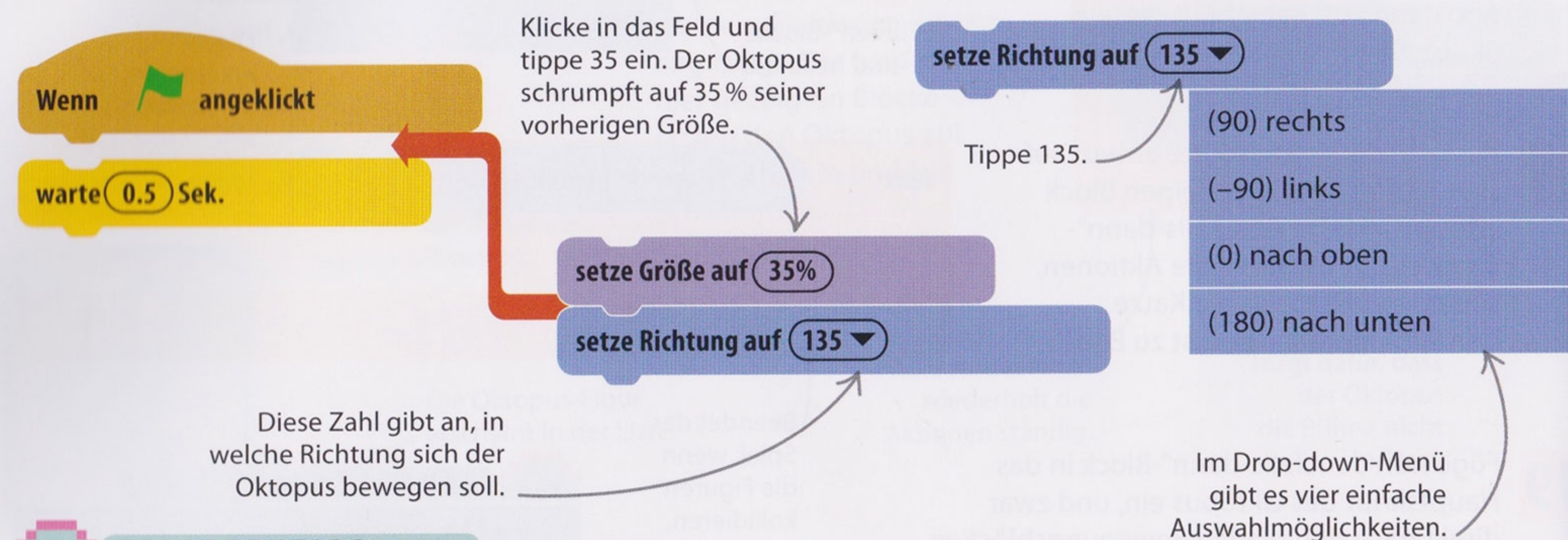
Täglich triffst du Entscheidungen. Wenn es regnet, nimmst du einen Regenschirm mit, wenn nicht, dann nicht. Computerprogramme brauchen dazu das, was Programmierer als Bedingungen bezeichnen, z. B. „falls-dann“. Trifft Scratch auf einen solchen Block, führt es die darin enthaltenen Blöcke nur aus, wenn die Aussage wahr ist.



Noch mehr Feinde

Wir fügen weitere Feinde hinzu. Besonders spannend wird es, wenn sie sich in verschiedene Richtungen bewegen. Damit jede Figur unabhängig gesteuert werden kann, gibt es einen Block, der wie ein Kompass funktioniert.

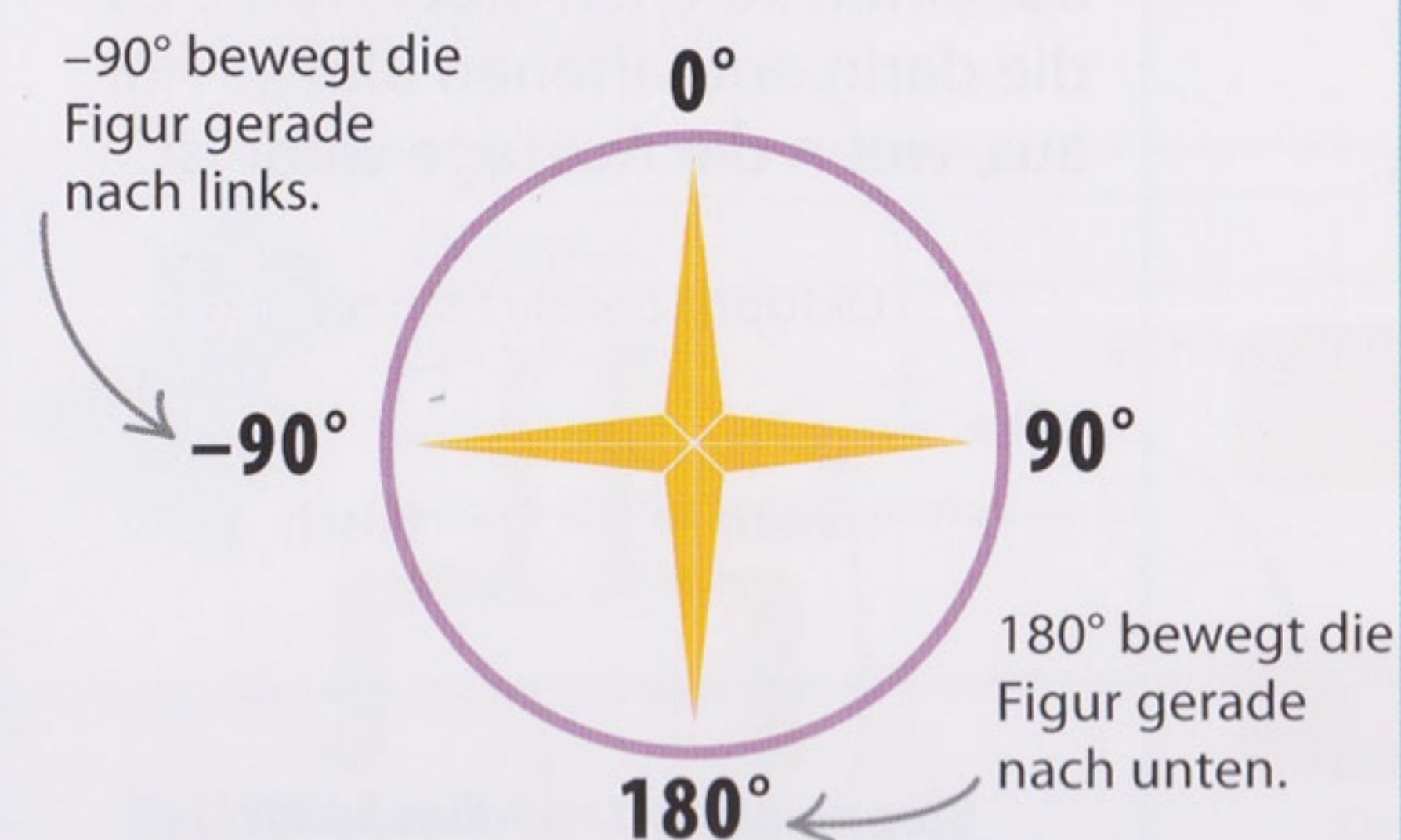
- 20** Füge direkt unter dem Block „Wenn angeklickt“ einen violetten Block „setze Größe auf“ in das Skript des Oktopus ein und setze die Größe auf 35 %. Füge dann noch einen Block „setze Richtung auf“ ein.



EXPERTENTIPP

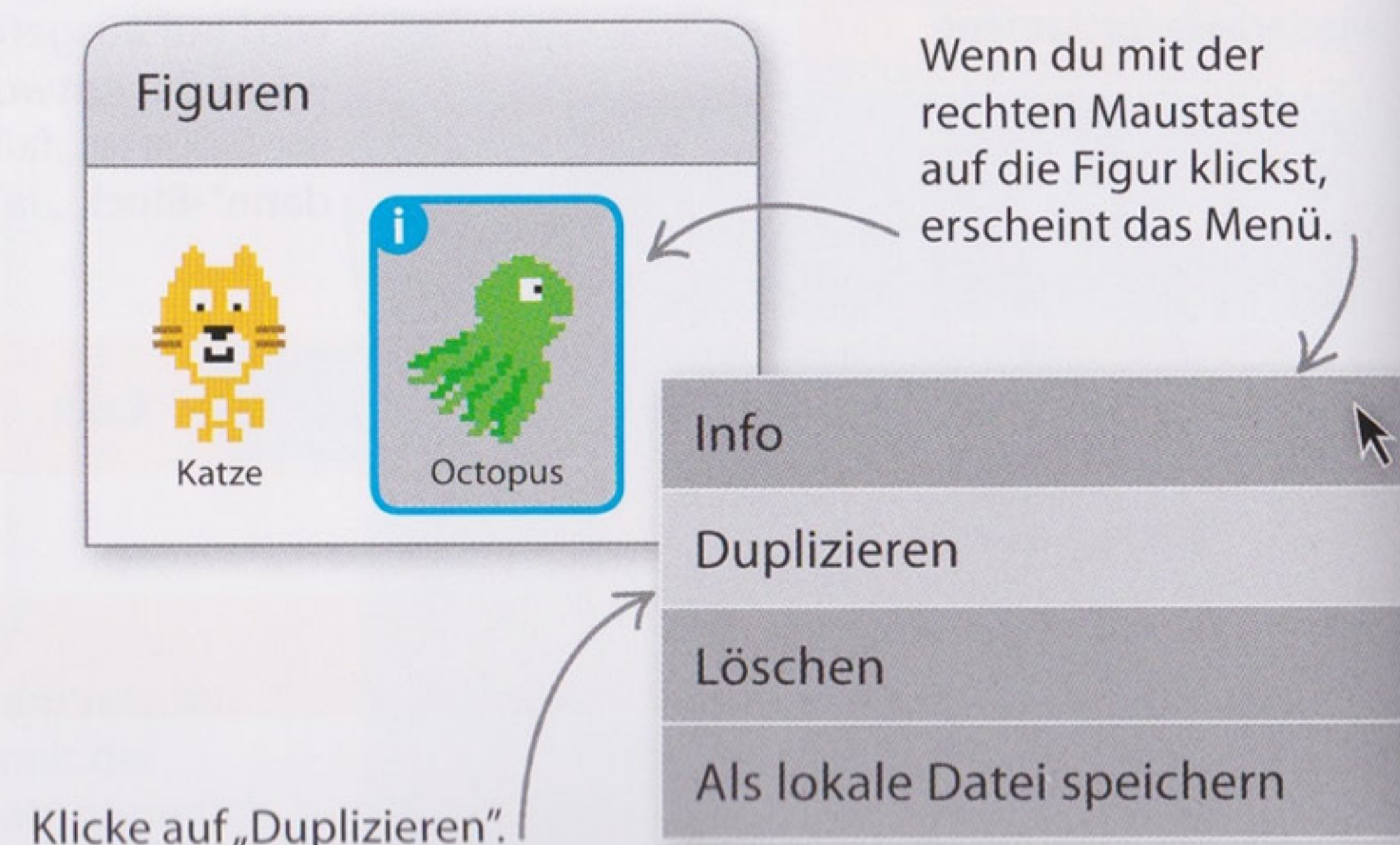
Richtungen

Scratch gibt Richtungen in Winkelgraden an. Jede Zahl von -179° bis 180° ist möglich. Mit negativen Zahlen bewegen sich die Figuren nach links, mit positiven nach rechts. 0° ist nach oben, 180° nach unten.



- 21** Um die Richtung des Oktopus zu ändern, klicke im Block „setze Richtung auf“ in das Feld mit der Zahl. Ersetze die 90 z. B. durch 135. Der Oktopus bewegt sich ab jetzt diagonal.

- 22** Wir duplizieren den Oktopus. Klicke mit der rechten Maustaste auf den Oktopus (beim Mac drücke „Strg“ und klicke) und wähle „Duplizieren“ aus. Wiederholst du das Ganze zweimal, so erscheinen in der Figurenliste Kopien mit den Namen „Octopus2“ und „Octopus3“. Jede besitzt eine Kopie des Oktopus-Skripts.



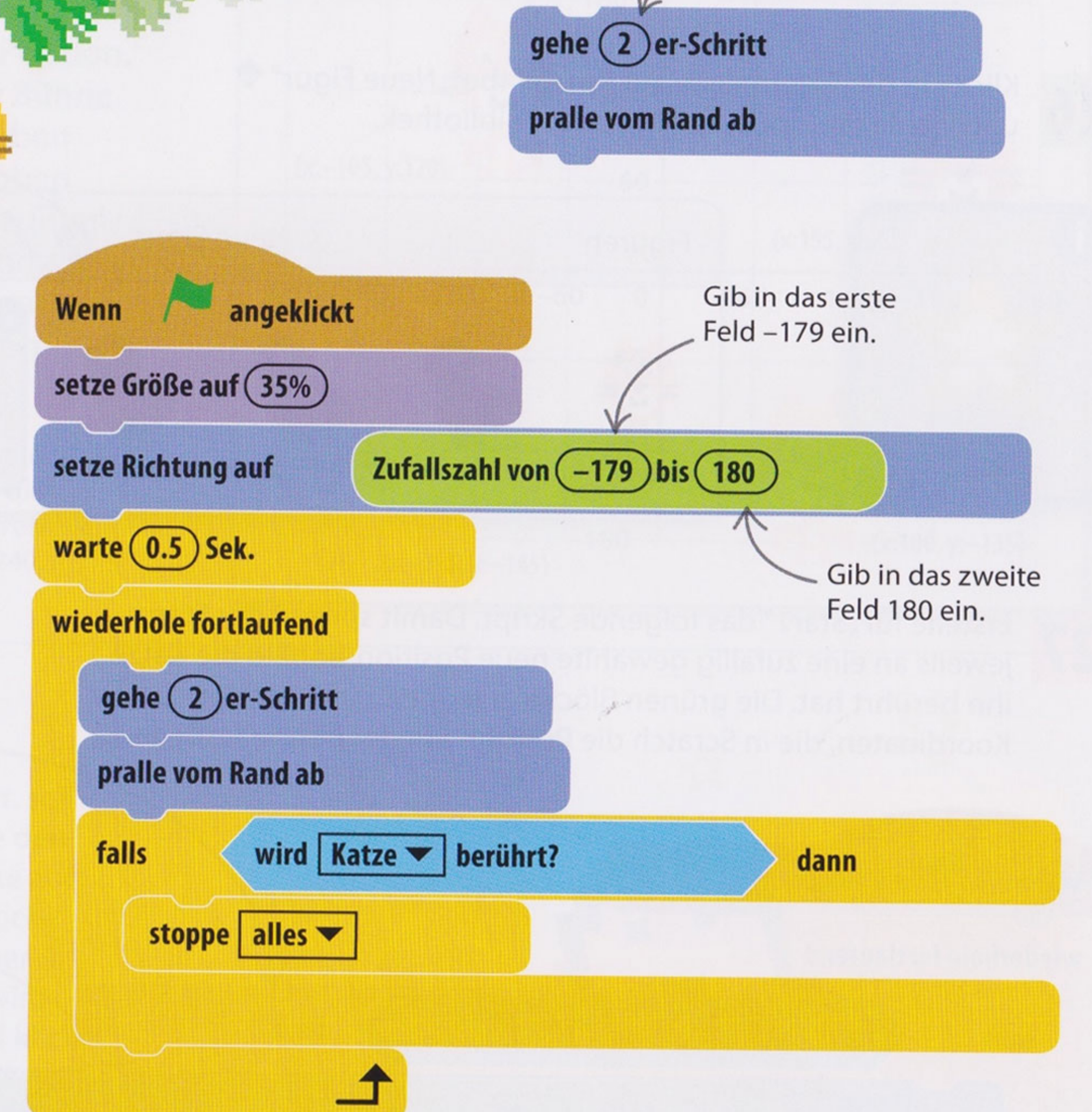
- 23** Jeder Oktopus muss eine eigene Zahl im Feld „setze Richtung auf“ haben, damit sie sich alle in unterschiedliche Richtungen bewegen. Setze daher „Octopus2“ auf „0“ und „Octopus3“ auf „90“. Führe das Programm aus und versuche nun, den drei Feinden auszuweichen.



- 25** Noch mehr Abwechslung erhalten wir, wenn einer der Oktopusse sich in eine zufällig gewählte Richtung bewegt. Dazu setzen wir den grünen Block „Zufallszahl“ ein. Scratch erzeugt damit, ähnlich wie ein Würfel, eine zufällige Zahl. Du findest den Block unter „Operatoren“ in der Blockpalette. Füge ihn in das Skript des ersten Oktopus ein. Wenn du das Programm mehrmals ausführst, siehst du, dass der Oktopus nun anfangs jeweils in eine andere Richtung startet.

- 24** Wenn es dir schwerfällt, den Feinden auszuweichen, verringere die Zahl der Schritte im Block „Bewegung“ auf „2“. Denke daran, das Skript jeder einzelnen Oktopus-Figur zu ändern.

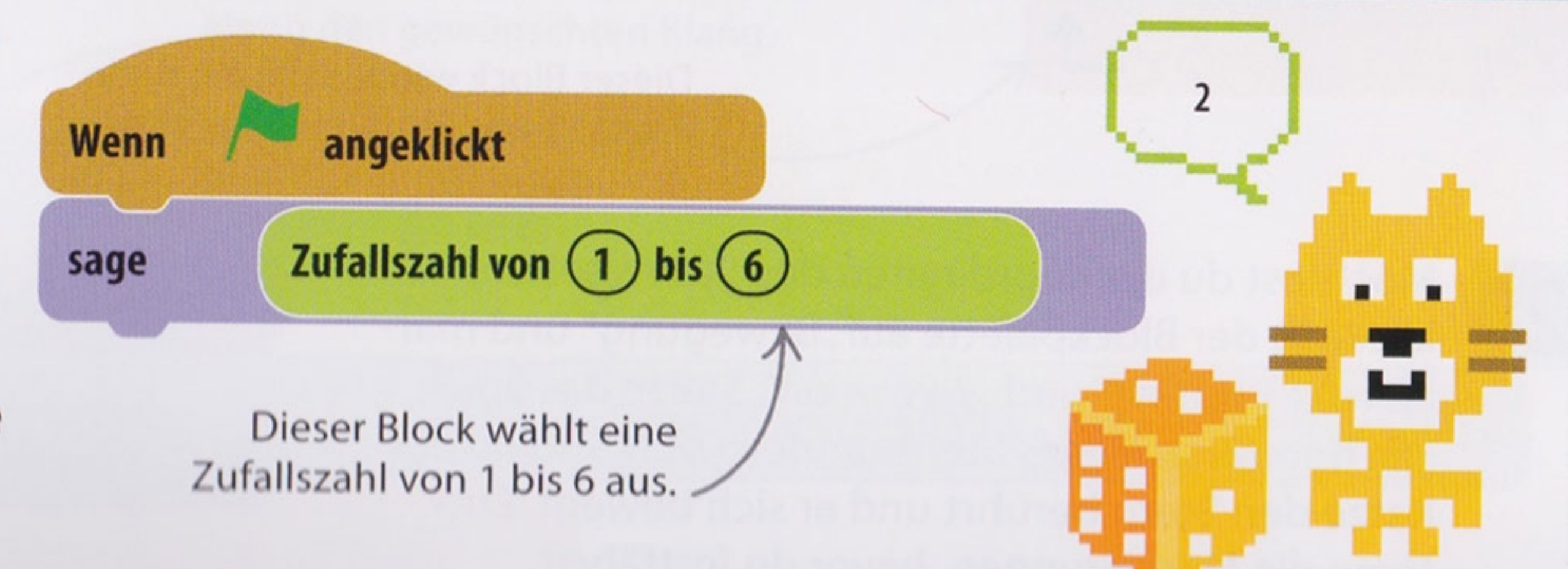
Diese Zahl legt die Geschwindigkeit des Oktopus fest.



EXPERTENTIPP

Zufallszahlen

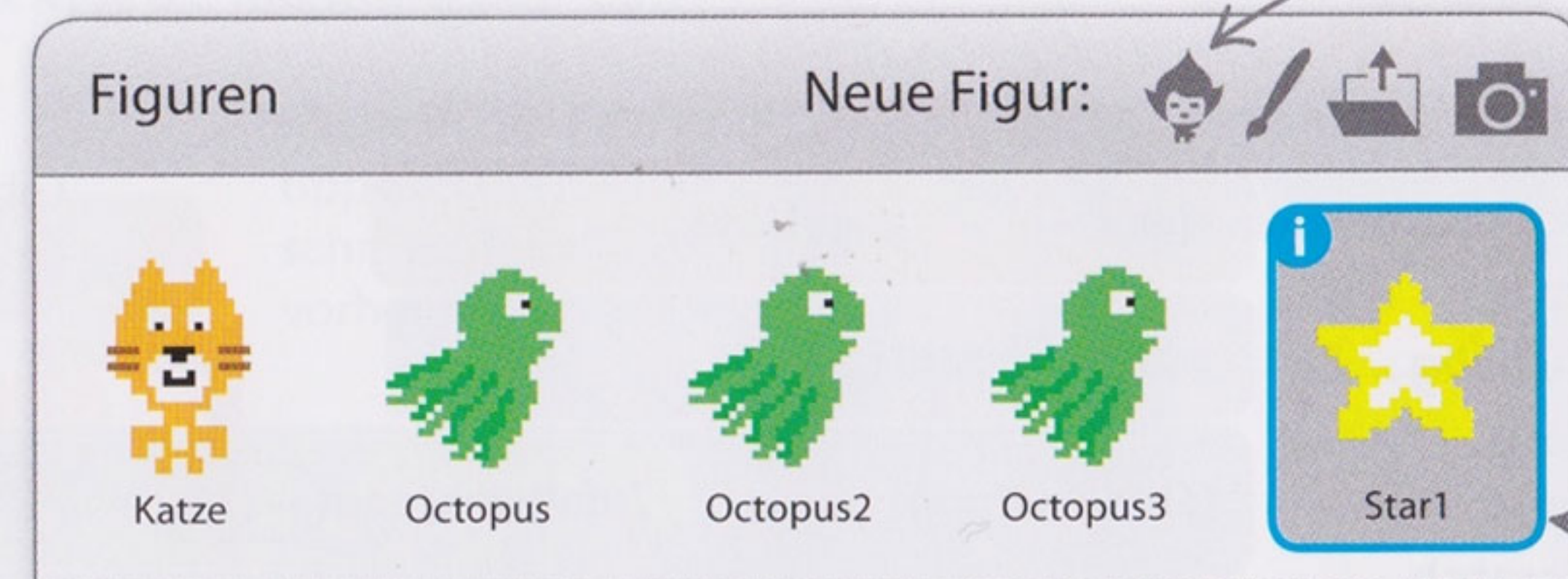
Warum gibt es so viele Würfelspiele? Würfel machen Spiele unberechenbar, weil sie dafür sorgen, dass den Spielern unvorhersehbare Dinge passieren. Auch Zufallszahlen lassen sich nicht vorhersagen. Du kannst die Katze so programmieren, dass sie eine zufällige Zahl zwischen 1 und 6 nennt.



Sterne sammeln

In vielen Spielen müssen wertvolle Gegenstände gesammelt werden, um Punkte zu erzielen oder am Leben zu bleiben. In Sternsammler muss der Spieler einen Schatz aus goldenen Sternen sammeln. Jeder Stern erscheint an einem zufälligen Ort.

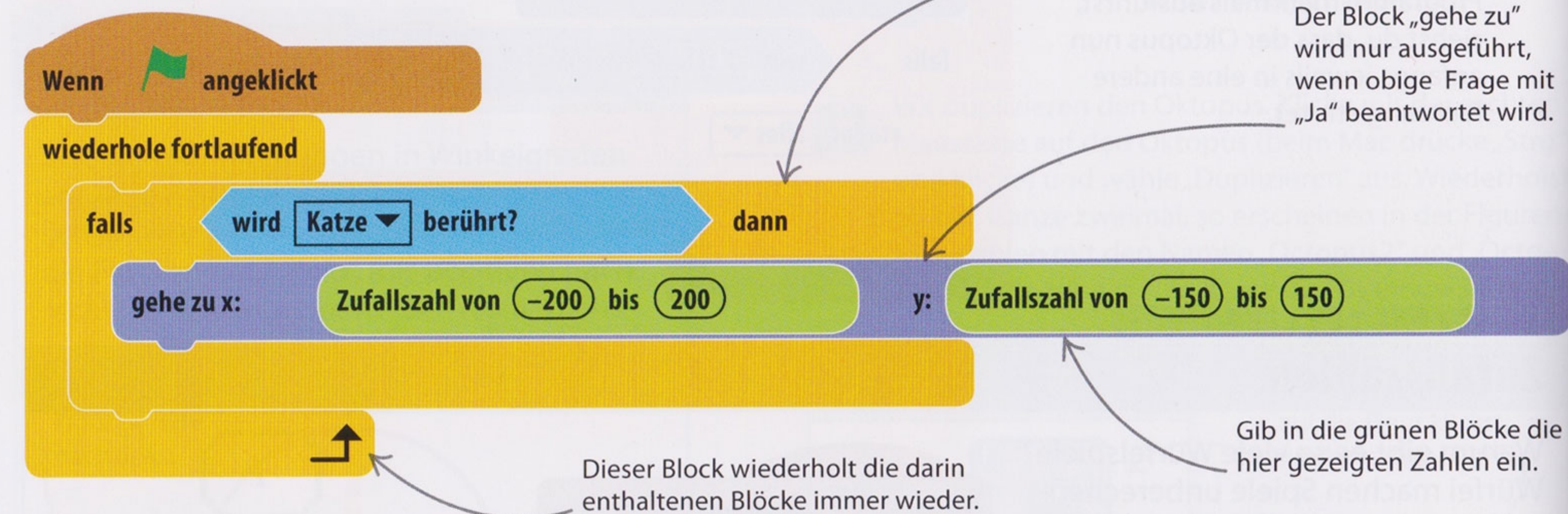
- 26** Klicke in der Figurenliste auf das Symbol „Neue Figur“ und wähle die Figur „Star1“ aus der Bibliothek.



Ein Klick auf dieses Symbol öffnet die Figurenbibliothek.

Die Figur „Star1“ wird in der Figurenliste angezeigt.

- 27** Erstelle für „Star1“ das folgende Skript. Damit springt der Stern jeweils an eine zufällig gewählte neue Position, sobald die Katze ihn berührt hat. Die grünen Blöcke erzeugen Zufallszahlen für die Koordinaten, die in Scratch die Position auf der Bühne festlegen.



- 28** Möchtest du die Koordinaten des Sterns sehen, klicke in der Blockpalette auf „Bewegung“ und markiere „x-Position“ und „y-Position“. Starte das Spiel: Die Koordinaten des Sterns ändern sich, sobald die Katze den Stern berührt und er sich bewegt. Entferne die Markierungen, bevor du fortfährst.

Star1: x-Position (60)

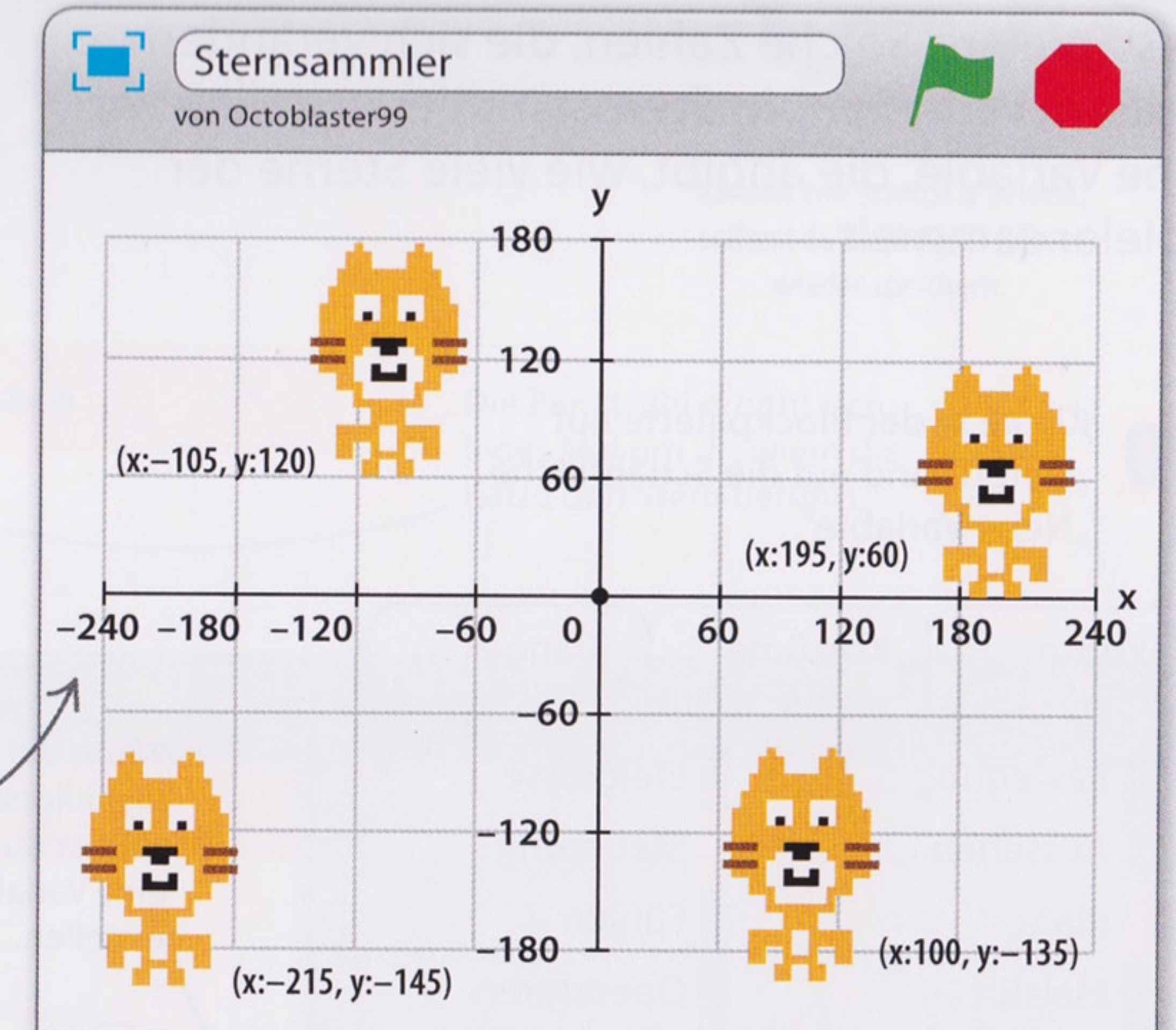
Star1: y-Position (78)

EXPERTENTIPP

Mit Koordinaten arbeiten

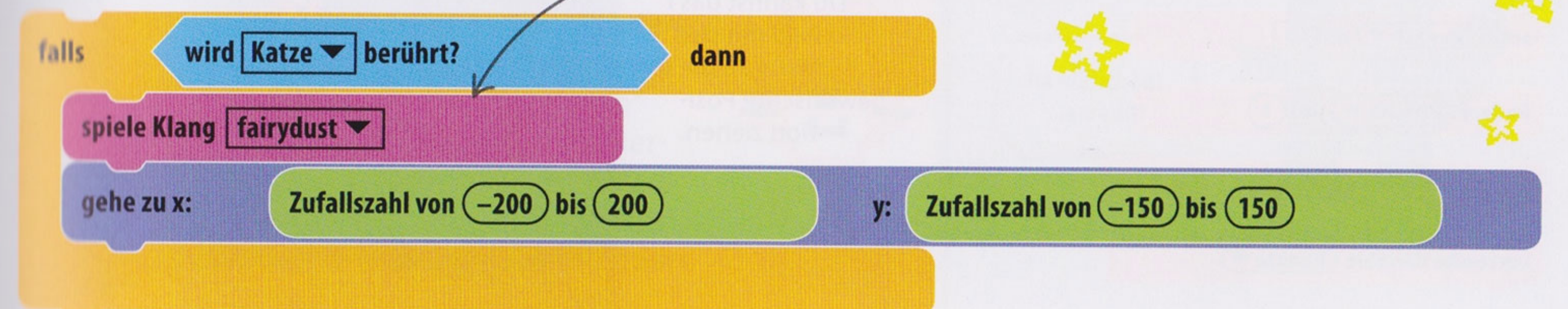
Die Position jeder Figur auf der Bühne wird in Scratch mit einem Zahlenpaar, sogenannten Koordinaten, angegeben. Wie bei mathematischen Graphen steht die x-Koordinate für die horizontale und die y-Koordinate für die vertikale Position. Gezählt wird vom Mittelpunkt der Bühne aus zu beiden Seiten sowie nach oben und unten. Positive Werte liegen oben oder rechts, negative Werte liegen unten oder links. Jeder Punkt auf der Bühne hat ein eindeutiges Koordinatenpaar, mit dem eine Figur an diese Position versetzt werden kann.

Die x-Achse ist in Scratch länger als die y-Achse. Sie reicht von -240 bis 240.



- 29** Wenn die Katze den Stern berührt, soll ein Klingeffekt ertönen. Markiere den Stern in der Figurenliste und klicke auf das Register „Klänge“ über der Blockpalette. Klicke auf das Lautsprecher-Symbol. Die Klangbibliothek wird geöffnet. Markiere „fairydust“ und klicke auf „OK“. Füge den Block „spiele Klang“ in das Skript des Sterns ein und wähle „fairydust“ in der Drop-down-Liste.

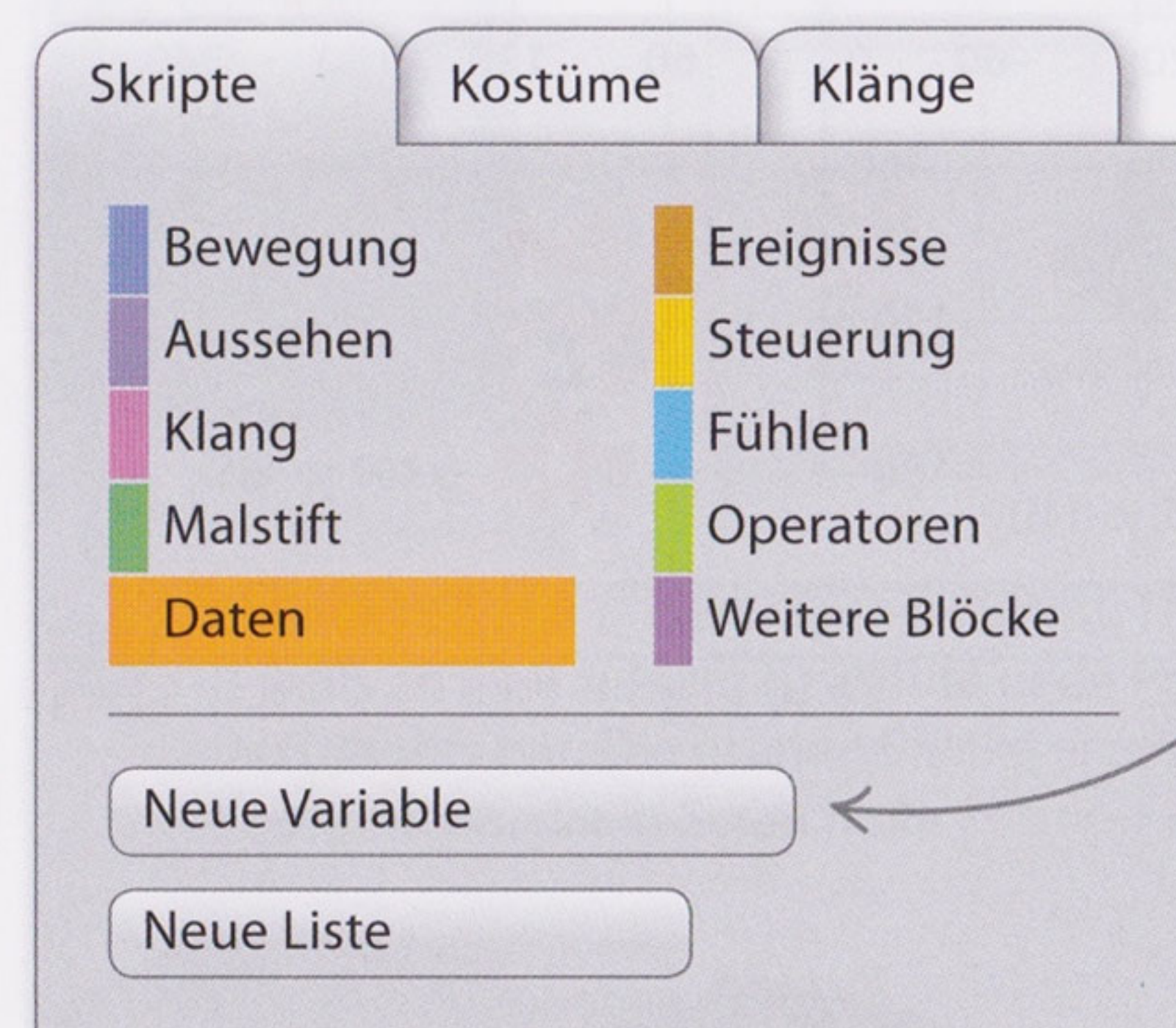
Füge den Block „spiele Klang“ in das Skript von „Star1“ ein und markiere dann im Drop-down-Menü den gewünschten Klang.



Punkte zählen

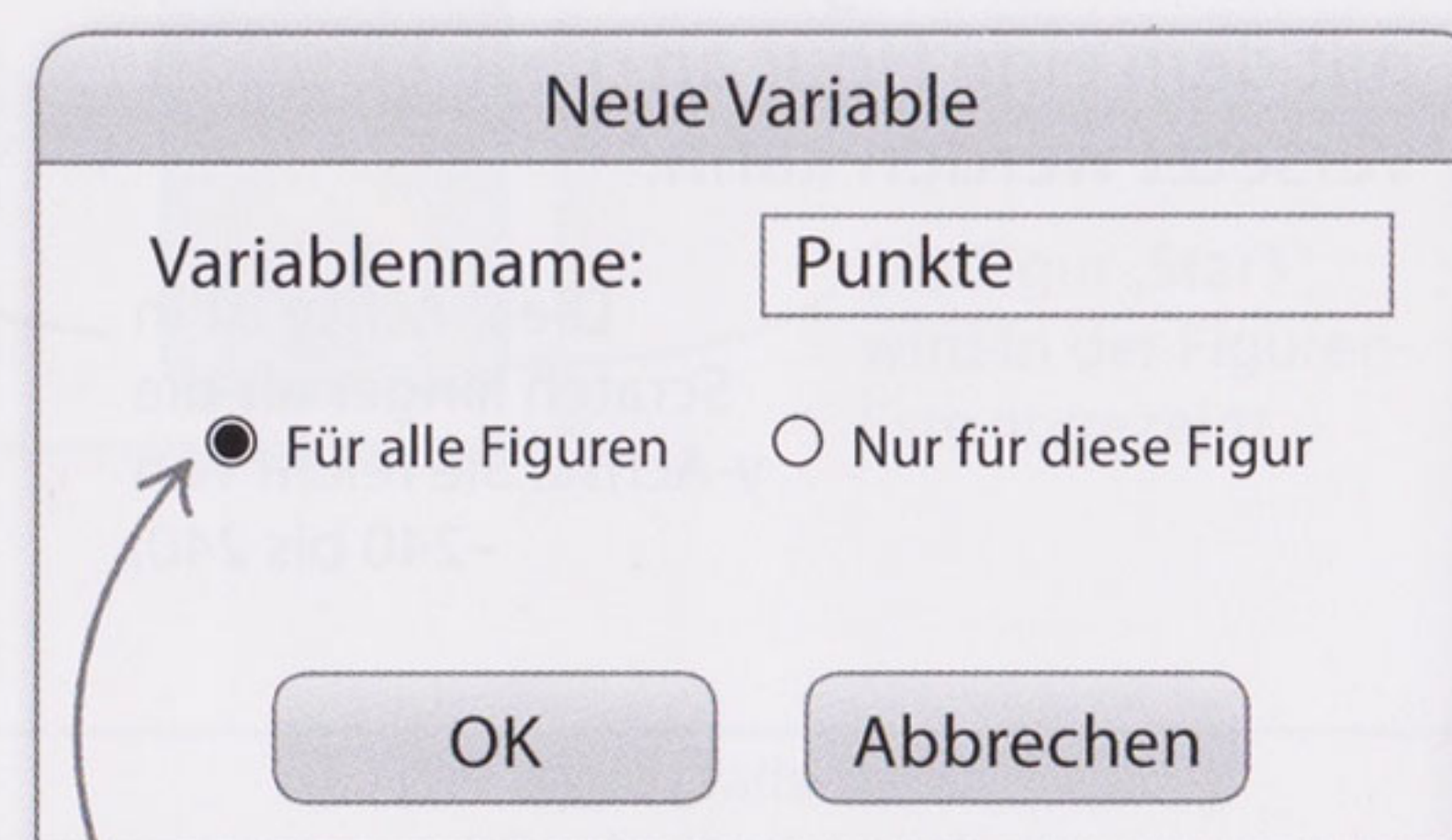
In Computerspielen werden wichtige Werte meist mitgezählt, z. B. die Punktzahl oder die Lebensdauer des Spielers. Solche Zahlen, die sich verändern, heißen „Variablen“. In Sternsammler erstellen wir eine Variable, die angibt, wie viele Sterne der Spieler sammelt.

30 Klicke in der Blockpalette auf „Daten“ und auf die Schaltfläche „Neue Variable“.



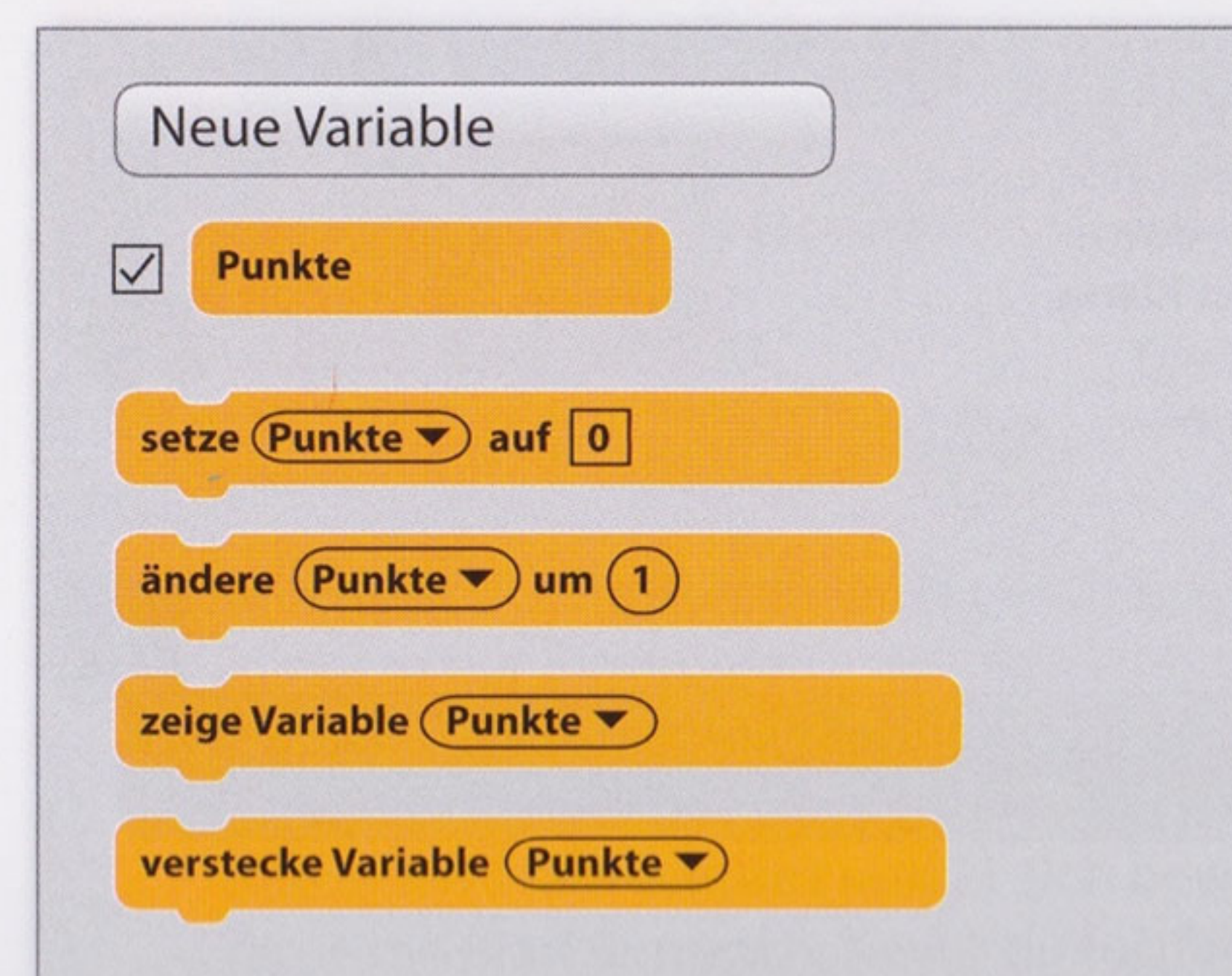
Wenn du hier klickst, kannst du eine Variable erstellen.

31 Tippe den Variablennamen „Punkte“ in das Feld. Prüfe, ob die Option „Für alle Figuren“ markiert ist, und klicke auf „OK“.



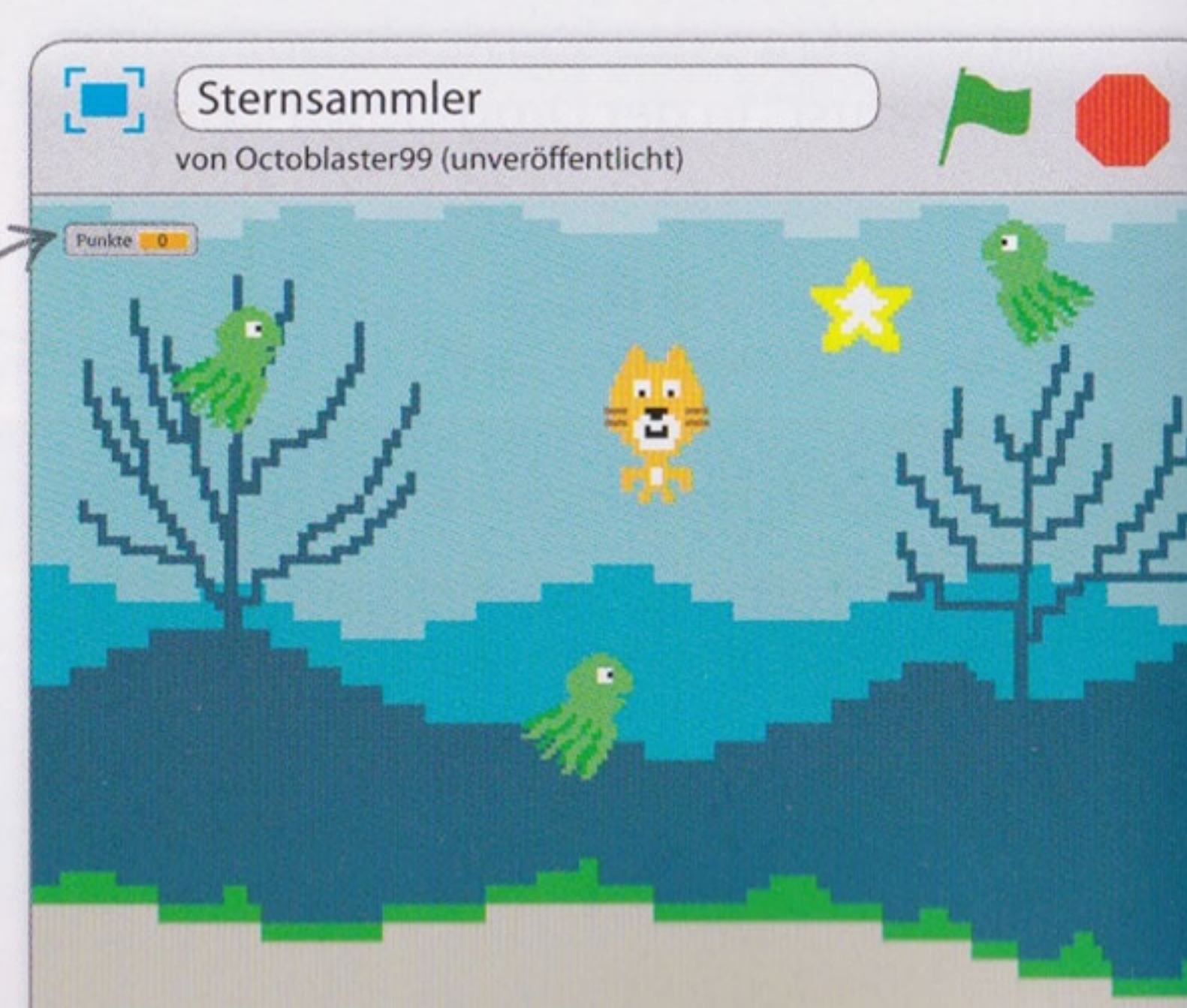
Mit dieser Option steht die Variable für alle Figuren zur Verfügung.

32 Einige neue Blöcke werden angezeigt, darunter einer für die Punktzahl. Das Kästchen muss markiert sein, damit die Punktzahl angezeigt wird.

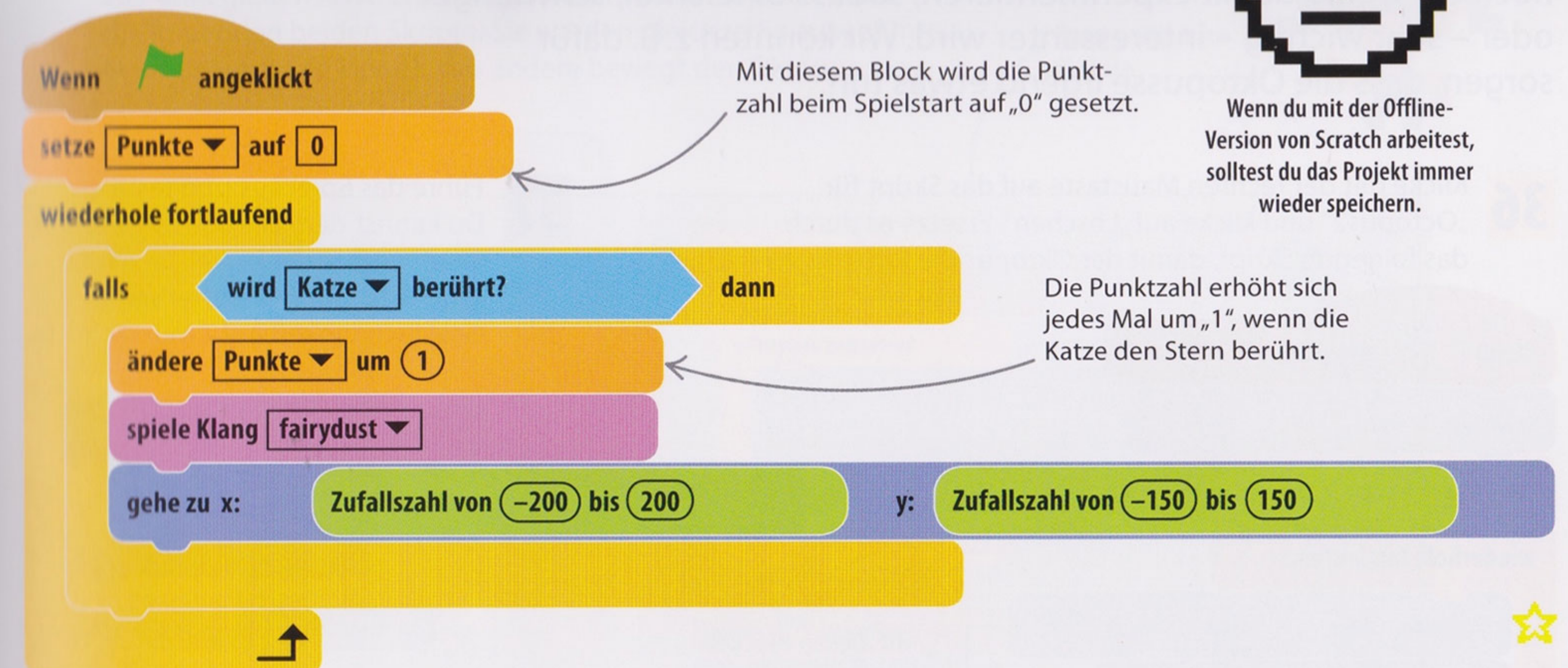


Du kannst das Feld mit der Maus an die gewünschte Position ziehen.

33 Das Feld für die Punktzahl erscheint nun auf der Bühne, du kannst es an die gewünschte Position ziehen.



34 Die Punktzahl soll bei Null beginnen und sich bei jeder Berührung eines Sterns um eins erhöhen. Markiere die Figur des Sterns und füge folgende zwei „Daten“-Blöcke in das Skript ein.



Wenn du mit der Offline-Version von Scratch arbeitest, solltest du das Projekt immer wieder speichern.

Die Punktzahl erhöht sich jedes Mal um „1“, wenn die Katze den Stern berührt.

35 Klicke auf die grüne Fahne und prüfe, was geschieht, wenn die Katze einen Stern sammelt. Schaffst du 20 Sterne, ohne mit einem Oktopus zusammenzustoßen?



EXPERTENTIPP Variablen

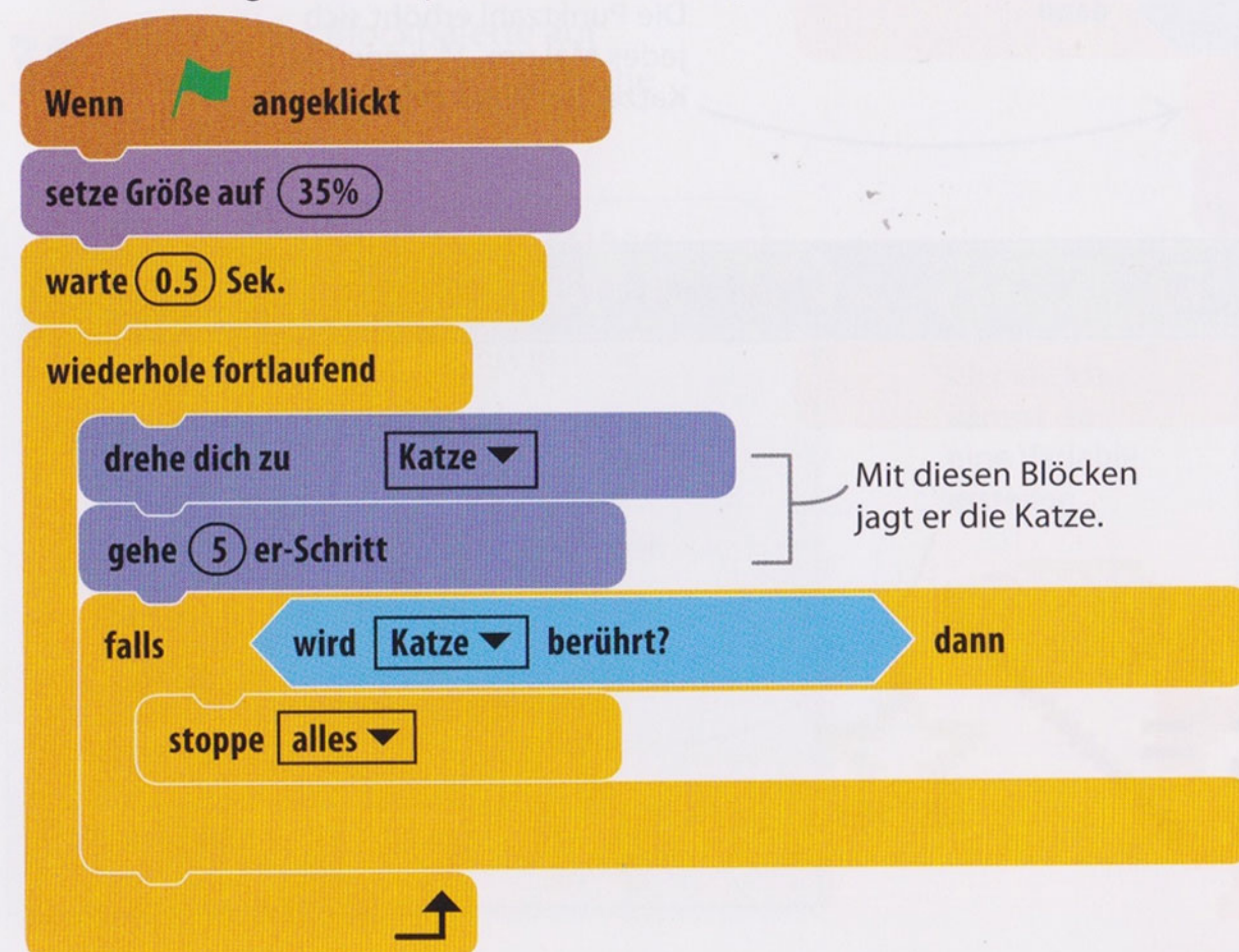
Eine Variable funktioniert wie eine Schachtel, in der Informationen aufbewahrt werden, z. B. eine Zahl, die sich ändert. In der Mathematik setzt man für Variablen Buchstaben wie x und y ein, in der Programmierung geben wir ihnen dagegen Namen wie „Punkte“. Wir speichern damit nicht nur Zahlen, sondern auch andere Informationen. Der Name sollte immer angegeben, wofür die Variable steht, z. B. „Tempo“ oder „Punkte“. Da die meisten Programmiersprachen keine Leerzeichen in Namen erlauben, musst du z. B. statt „Tempo des Hundes“ einen Namen wie „HundeTempo“ verwenden.



Bessere Feinde

Nun funktioniert das Spiel bereits gut, und wir können es testen und noch ein wenig damit experimentieren, sodass es leichter, schwieriger oder – sehr wichtig – interessanter wird. Wir könnten z. B. dafür sorgen, dass die Oktopusse irgend etwas tun.

36 Klicke mit der rechten Maustaste auf das Skript für „Octopus2“ und klicke auf „Löschen“. Ersetze es durch das folgende Skript, damit der Oktopus die Katze jagt.



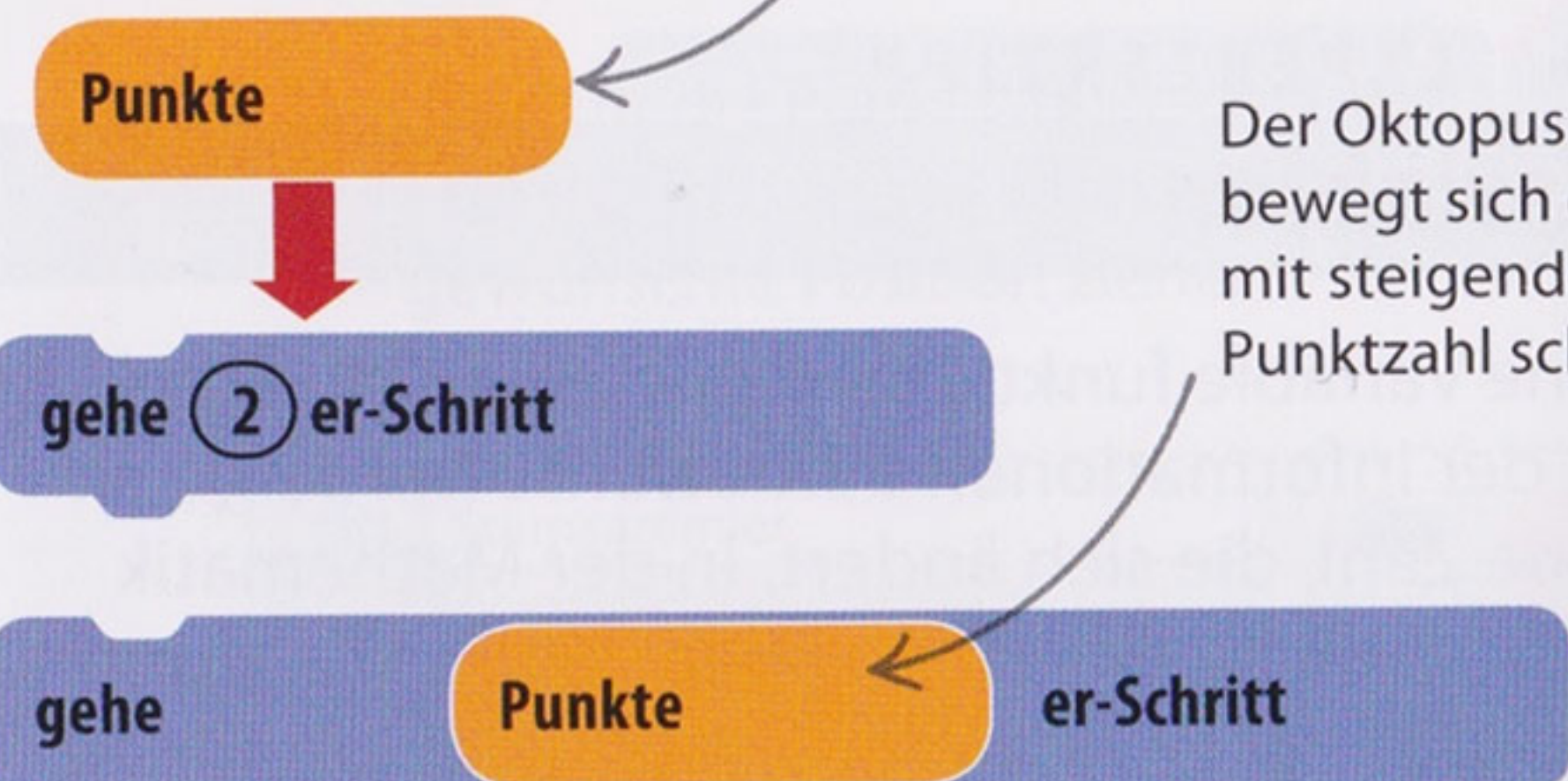
Mit diesen Blöcken jagt er die Katze.

37 Führe das Spiel aus und teste es. Du kannst dem Oktopus vermutlich kaum entkommen, weil er sehr schnell ist. Ändere daher seine Geschwindigkeit auf Zweierschritte.



Die Zahl steuert die Geschwindigkeit des Oktopus.

Setze den Block „Punkte“ in das kreisförmige Feld des Bewegungsblocks.



Der Oktopus bewegt sich dann mit steigender Punktzahl schneller.

38 Nach und nach wirst du besser, dann kannst du das Spiel schwieriger gestalten: Markiere die Original-Oktopusfigur und klicke in der Blockpalette auf „Daten“. Ziehe den Block „Punkte“ in den Bewegungsblock im Skript. Teste das Spiel. Je mehr Punkte du machst, desto schneller schwimmt der Oktopus.

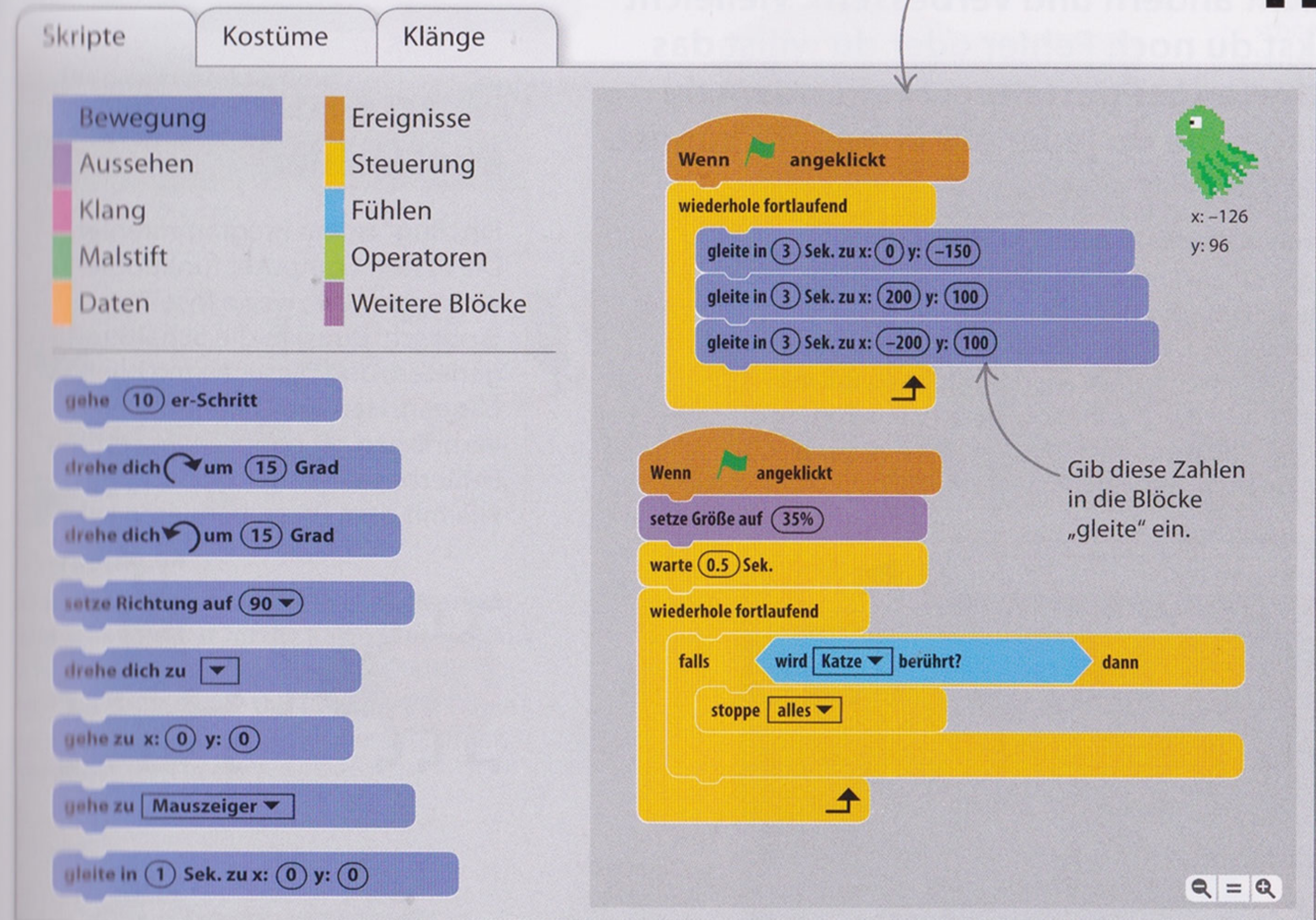
39 Ist das zu schnell, kannst du die Schwierigkeit auch langsamer steigern. Klicke in der Blockpalette auf „Operatoren“ und suche den kleinen Block „geteilt durch“. Ändere den Bewegungsblock nach dem hier gezeigten Vorbild. Gib in das runde Feld „3“ ein.

Der grüne Block teilt die Punktzahl durch 3, sodass der Oktopus sein Tempo langsamer steigert.



40 Nun lassen wir „Octopus3“ nach einem regelmäßigen Muster patrouillieren. Dazu brauchen wir einen neuen Bewegungsblock, der ihn nicht mehr in Schritten bewegt, sondern ihn sanft von Punkt zu Punkt gleiten lässt. Ersetze das Skript von „Octopus3“ durch die folgenden beiden Skripte. Sie werden gleichzeitig ausgeführt. Eines erkennt Kollisionen, das andere bewegt den Oktopus.

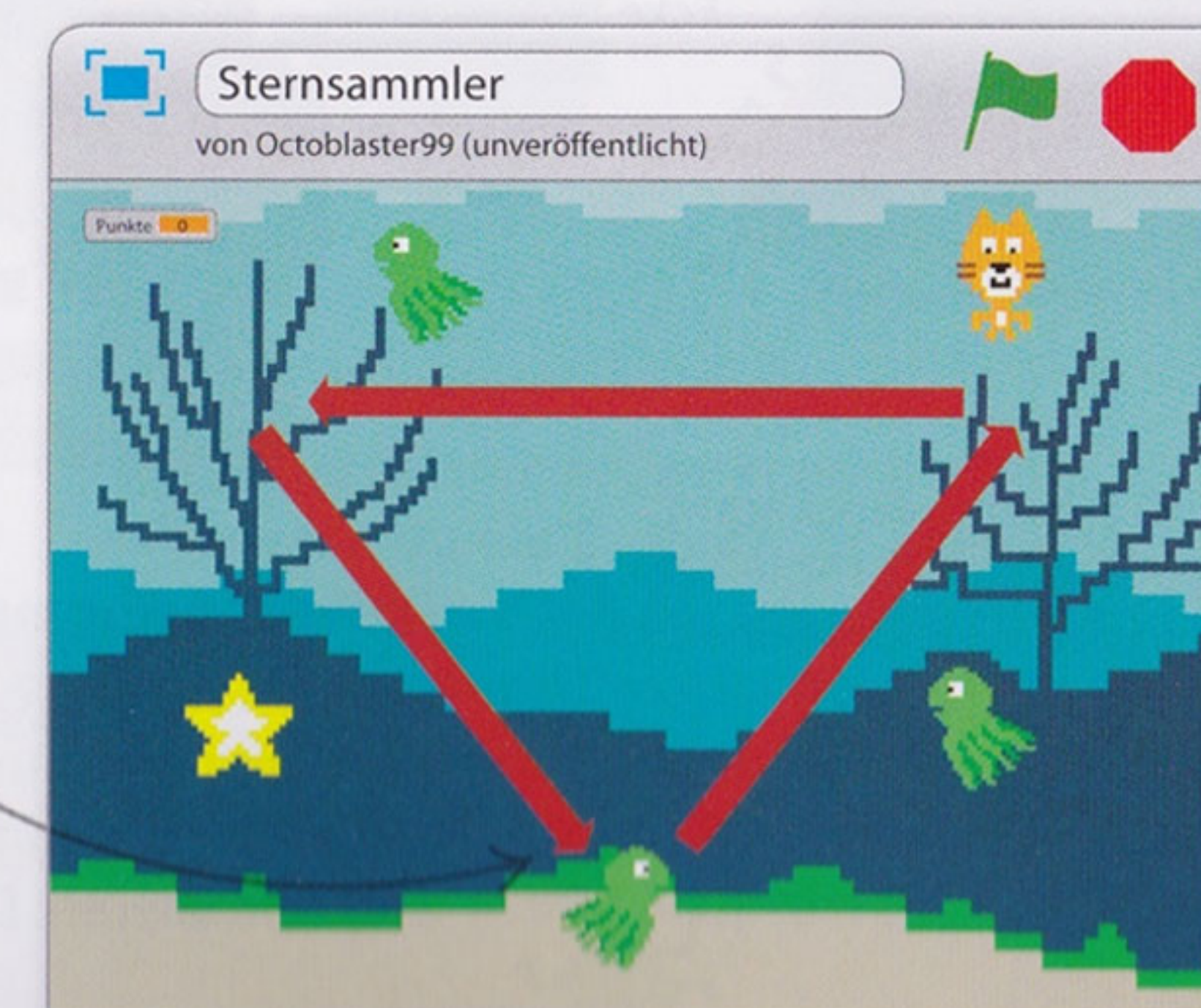
Zwei getrennte Skripte.



Gib diese Zahlen in die Blöcke „gleite“ ein.

41 Führe das Projekt aus und beobachte „Octopus3“. Er sollte in einem regelmäßigen Dreiecksmuster schwimmen.

Du kannst die Form des Dreiecks ändern, indem du andere Zahlen in die Blöcke „gleite“ eingibst.



Ich glaube, ich schwimme im Kreis ...

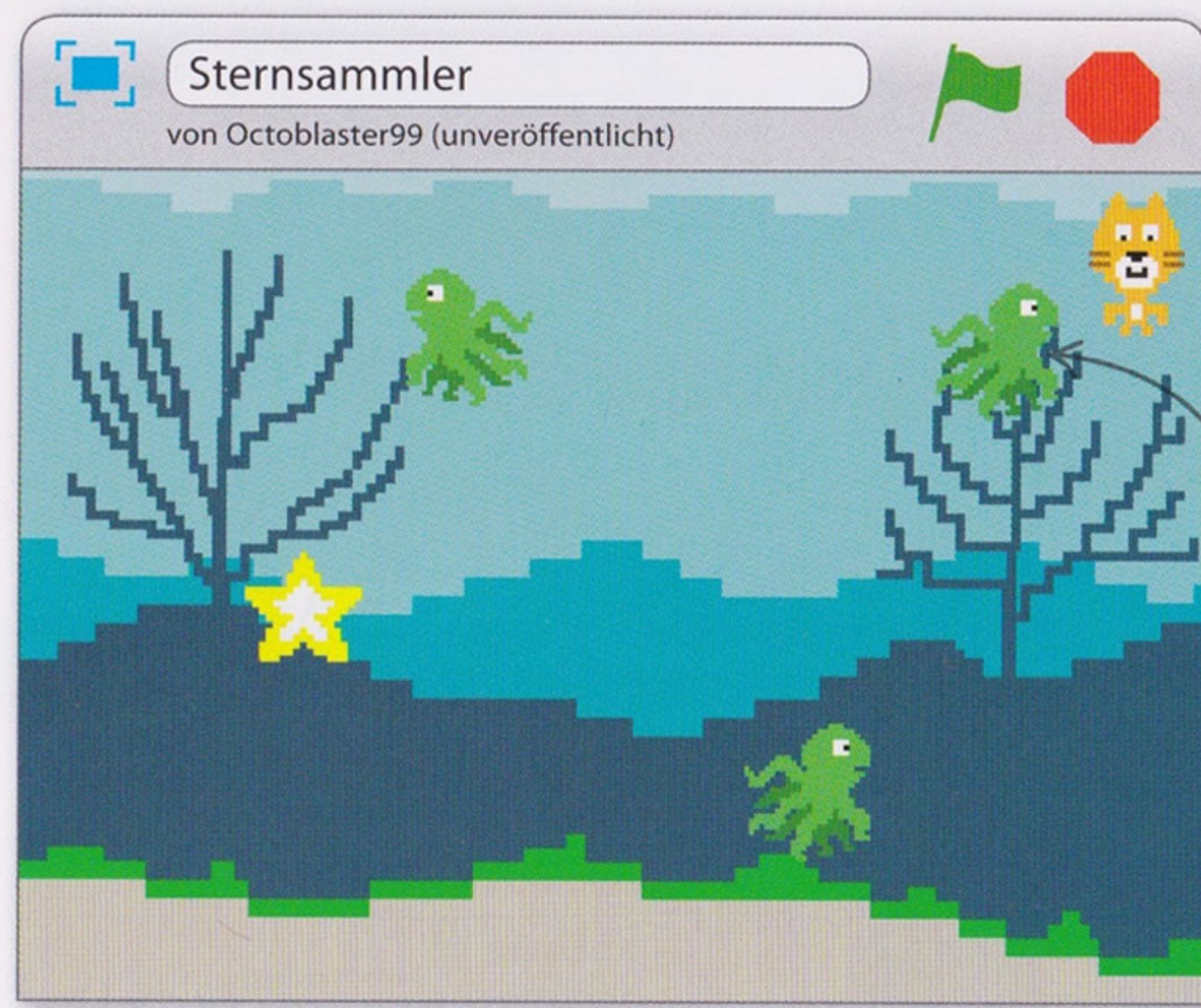


Der letzte Schliff

Du hast ein tolles Spiel programmiert, aber das ist erst der Anfang. In Scratch kannst du Spiele ganz leicht ändern und verbessern. Vielleicht entdeckst du noch Fehler oder du willst das Spiel schwieriger gestalten. Hier findest du Vorschläge, die du jederzeit erweitern kannst.

▽ Fehler von „Octopus2“ beheben (Debugging)

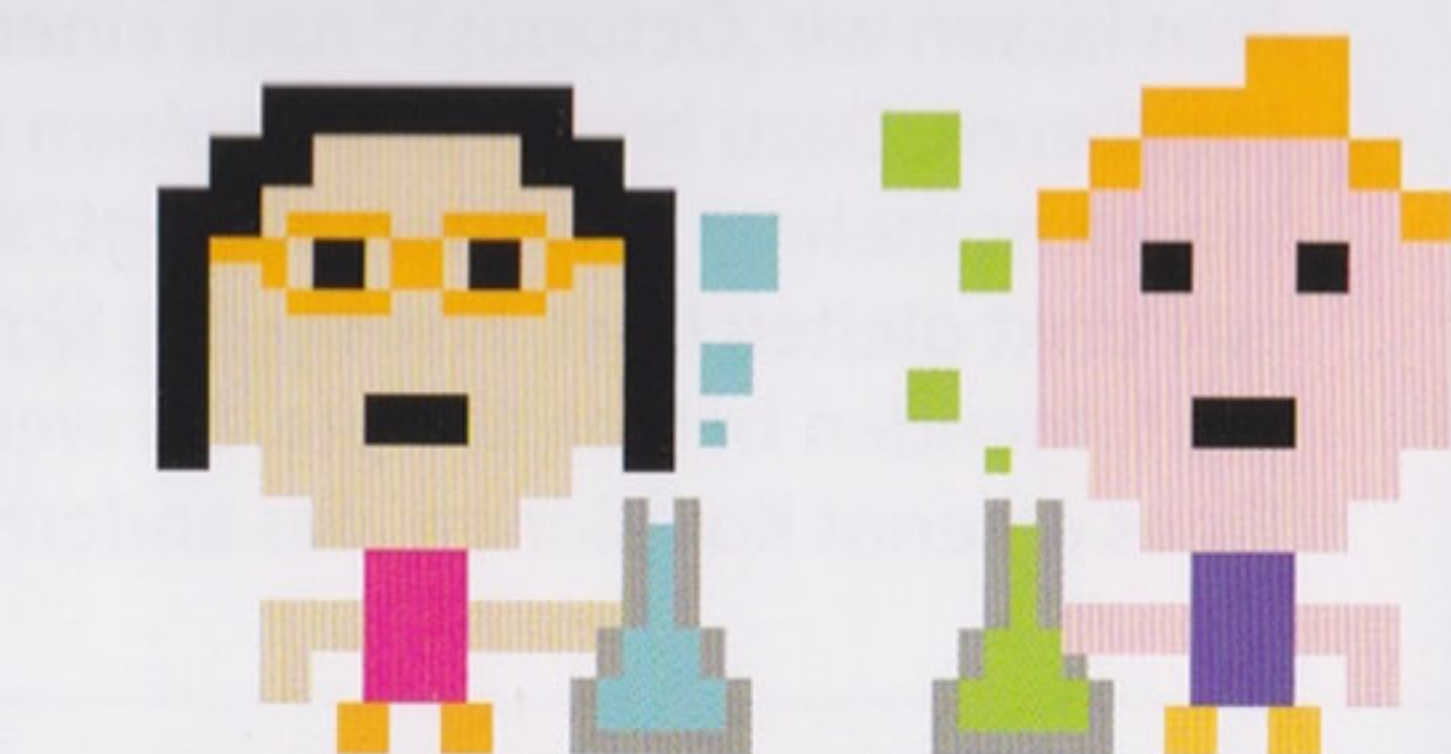
Bleibt „Octopus2“ am Ende eines Spiels in der rechten oberen Ecke stehen, sitzt der Spieler zu Beginn des nächsten Spiels fest. Das ist ein Fehler. Du kannst den Oktopus natürlich wegziehen, aber besser ist es, ihn mit einem Skript automatisch aus der Ecke zu holen. Füge zu Beginn des Skripts von „Octopus2“ den Block „gehe zu“ ein, der ihn in die Mitte der Bühne versetzt.



„Octopus2“ kann den Spieler in der rechten oberen Ecke festsetzen.

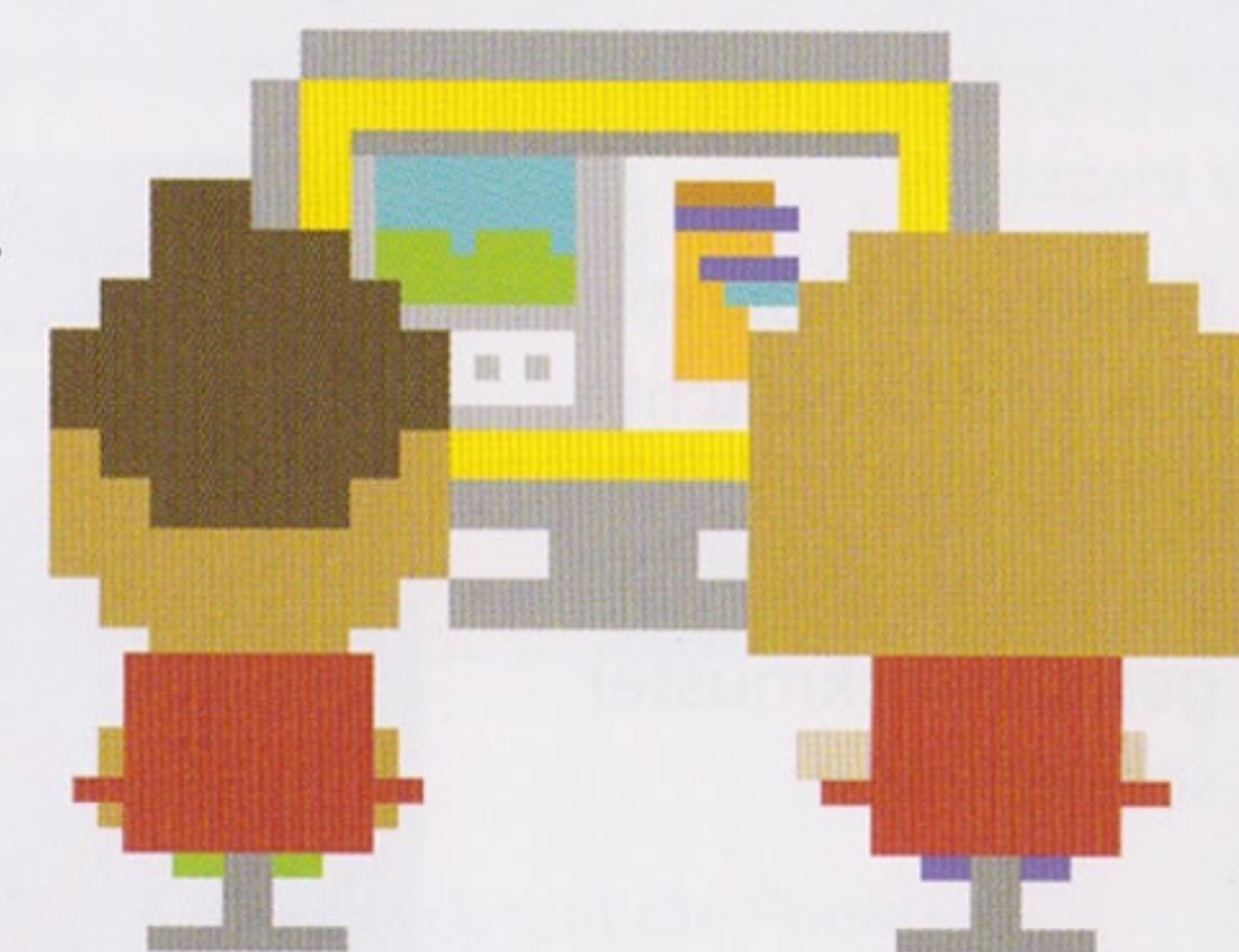
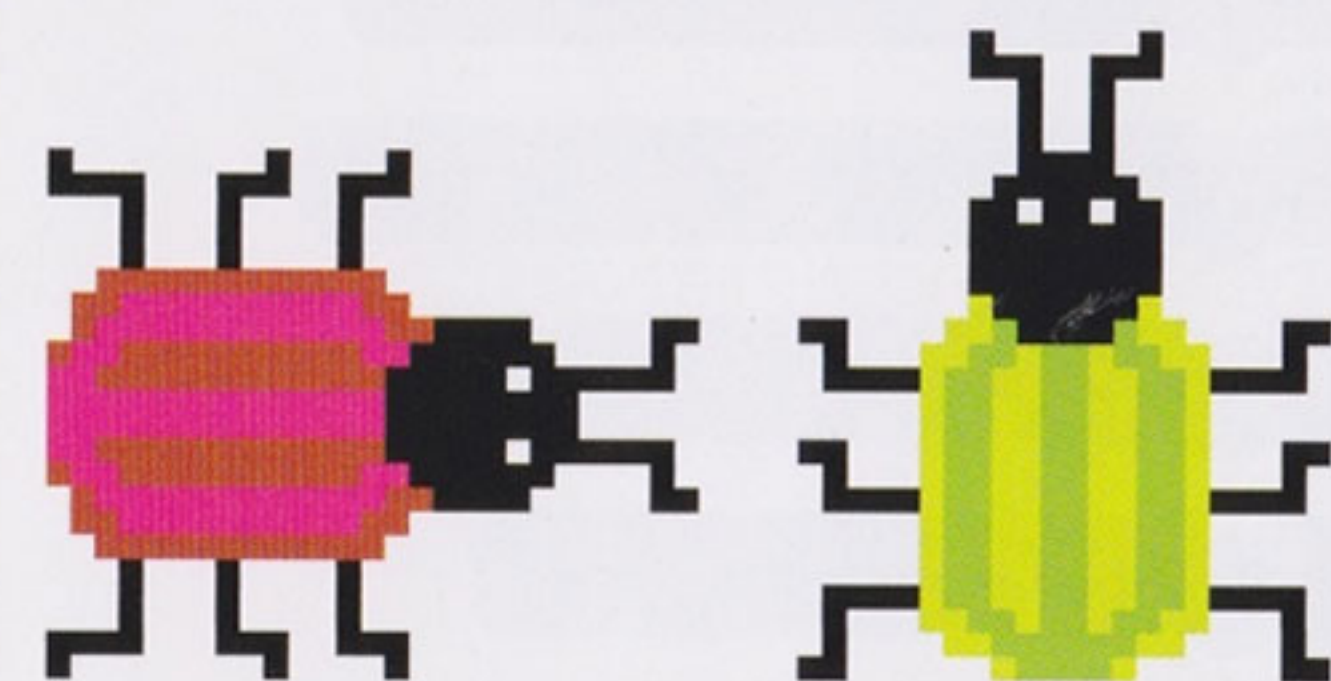


Mit diesem Block steht „Octopus2“ zu Spielbeginn in der Mitte der Bühne.



BEGRIFFE Bugs (Fehler)

Ein „Bug“ ist ein Programmfehler. Die ersten Computer funktionierten nicht mehr, wenn Insekten (Englisch: Bugs) in die Schaltkreise gerieten, und dieser Name blieb hängen. Heutige Programmierer verbringen oft ebenso viel Zeit mit Fehlerbeseitigung („Debugging“) wie mit dem Programmieren selbst.

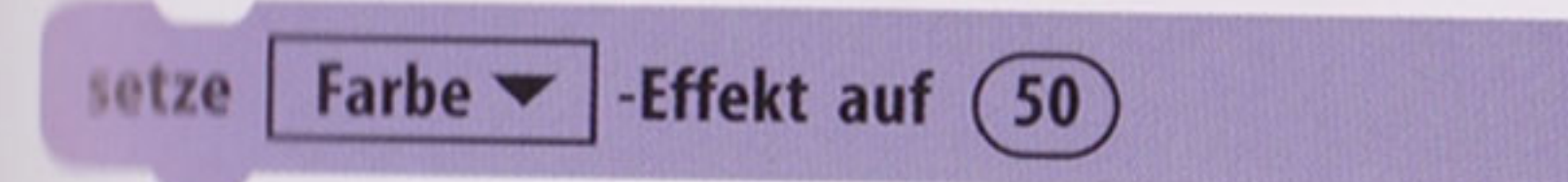


△ Feinabstimmung

Die besten Spiele werden sorgfältig getestet, damit alles funktioniert. Teste also jede Änderung und bitte auch Freunde, dein Spiel auszuprobieren: Dann weißt du, ob es gut ist.

▽ Verschiedene Farben

Die Oktopusse werden bunt, wenn du den Block „setze Farbe-Effekt auf“ aus dem Bereich „Aussehen“ unterhalb des Blocks „setze Größe-Effekt auf“ einfügst.

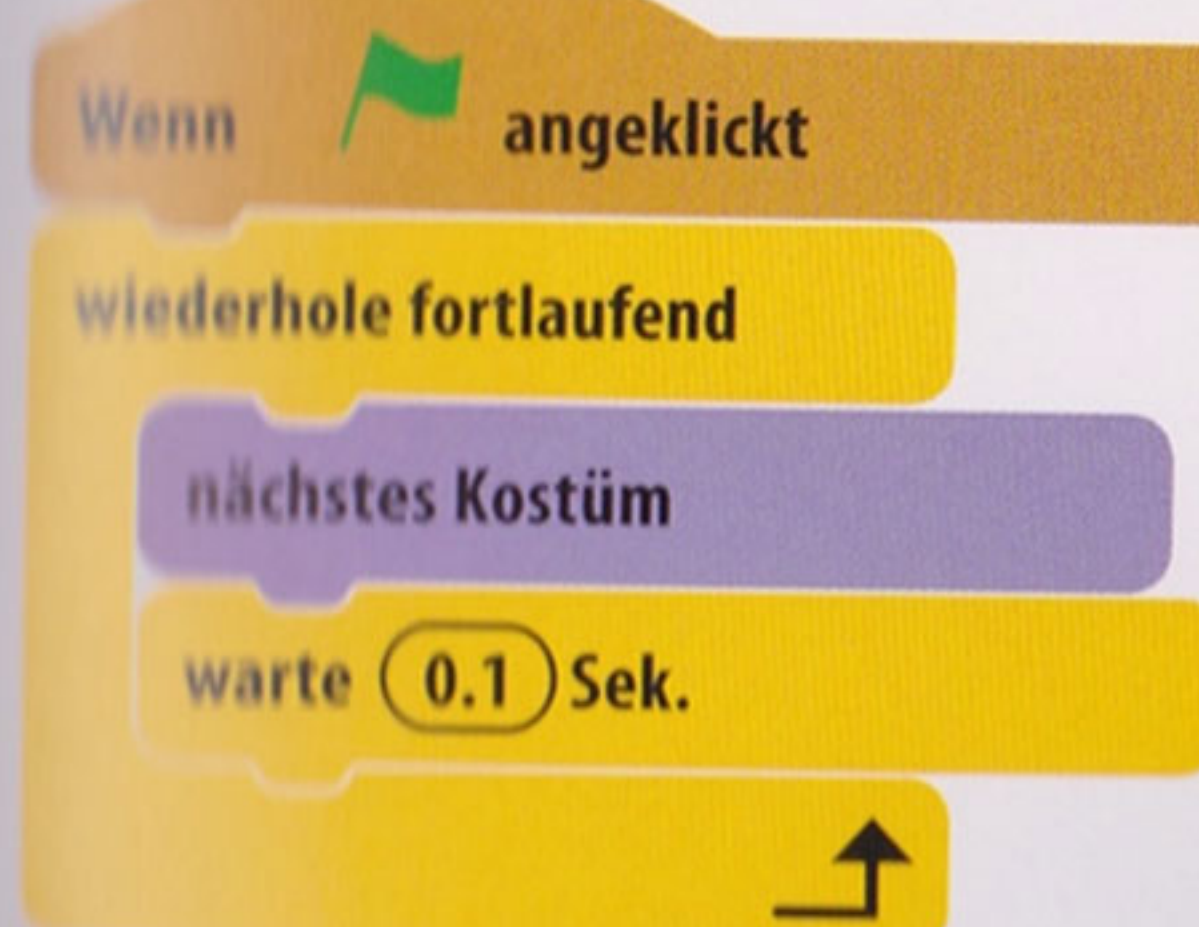


Die ganze Farbskala siehst du, wenn du hier verschiedene Zahlen zwischen -100 und 100 einsetzt.



△ Taucher

Damit das Unterwasser-Spiel überzeugender wirkt, kannst du die Katze in einen Taucher verwandeln. Klicke in der Figurenliste auf die Katze, öffne das Register „Kostüme“ und klicke auf . Lade aus der Bibliothek das Kostüm „Diver1“.



◁ Schwimm-Animation

„Sternsammler“ wirkt noch echter, wenn du die Tintenfische animierst. Sie sehen dann aus, als würden sie schwimmen. Das hier gezeigte Skript lässt die Tintenfische zwischen zwei Posen wechseln. Füge es für jeden Oktopus in den Skriptbereich ein.

▽ Farbenspiel

Ein Oktopus könnte die Farbe auch laufend ändern, sodass ein Blinkereffekt entsteht. Statte einen Oktopus mit folgendem Skript aus. Probiere im Block „ändere Farbe-Effekt um“ ruhig verschiedene Zahlen aus.

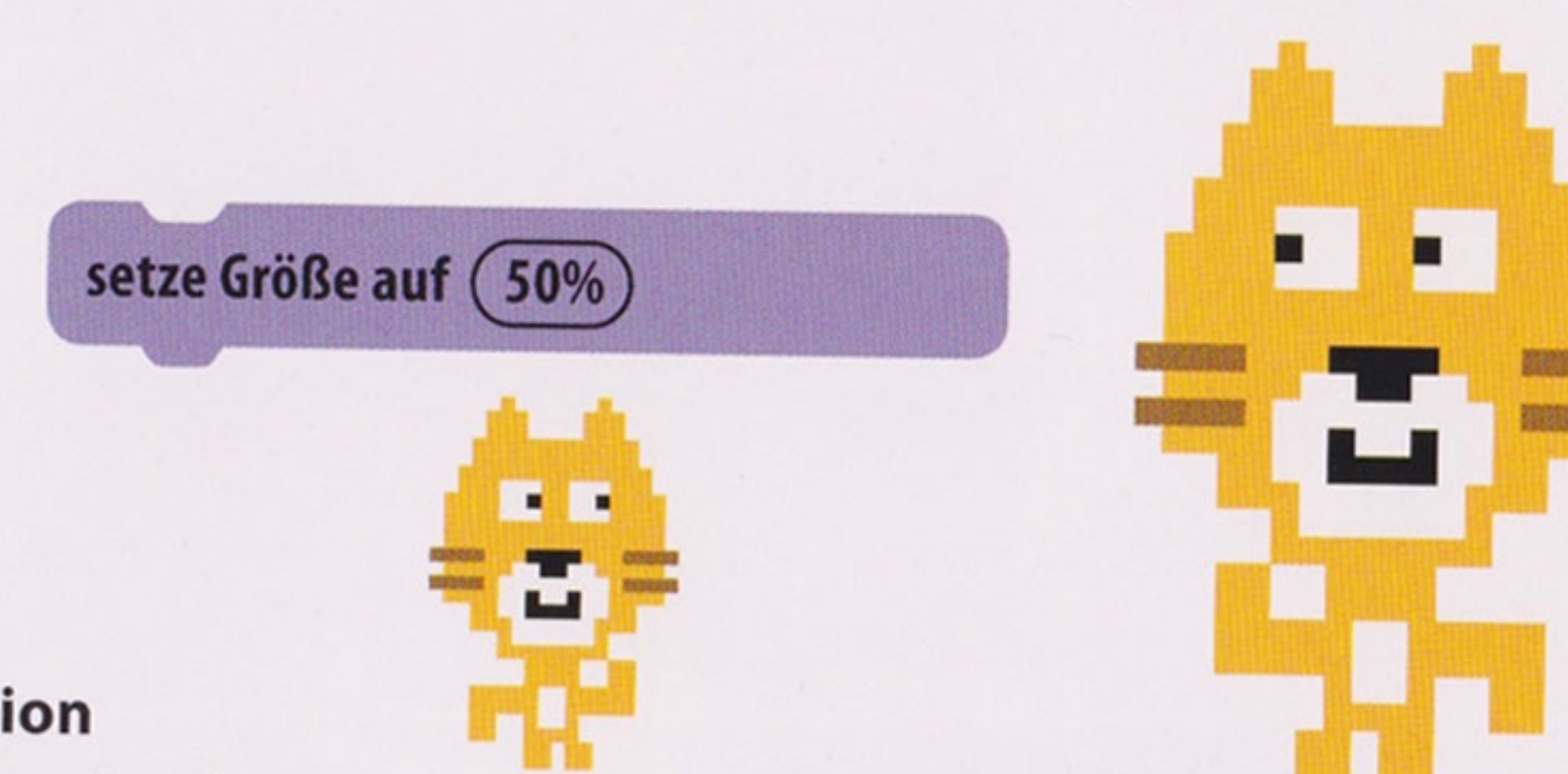


Diese Zahl bestimmt, wie schnell sich die Farben ändern.



▽ Verschiedene Größen

Du kannst den Schwierigkeitsgrad auch dadurch verändern, dass du die Figuren kleiner oder größer machst. Die Größe wird von der Zahl im Block „setze Größe auf“ festgelegt. Die Zahl in den blauen Bewegungsblöcken bestimmt dagegen ihr Tempo. Stelle die Zahlen so ein, dass du Freude am Spielen hast.



setze Größe auf 100%

