

## Werkzeuge für den Informatikunterricht

### ---- Scratch ----

#### 1. Kurzvorstellung

Scratch ist eine visuelle Programmiersprache für Kinder inklusive einer Entwicklungsumgebung und einer Online-Community-Plattform. Die Entwicklungsumgebung kann für Windows, Mac OS oder Linux heruntergeladen oder im Browser genutzt werden.

Mit Scratch kann man Geschichten, Spiele oder Animationen programmieren. Die angewendeten Programmierparadigmen sind dabei imperativ und ereignisorientiert, teilweise auch objektorientiert.

Ein Programm besteht aus Bühne(n) und Figuren, denen der Programmcode jeweils zugeordnet ist. Programmiert wird mittels Drag & Drop von vorgefertigten Blöcken (Ereignisse, Schleifen, Bewegungen, ...) und Variablen. Es können auch eigene Blöcke und Variablen erstellt werden.

#### 2. Einordnung in die Lehrpläne

| OS   | Gymnasium  | FOS Informatik  | Berufliches Gymnasium  |
|--|--|---|--|
| Klasse 8 - LB 2:<br>Informationen verarbeiten:<br>Modell - Algorithmus -<br>Lösung<br>- Programmstrukturen:<br>Folge, Wiederholung,<br>Verzweigung<br>- Übertragen der Kennt-<br>nisse zum Problemlöse-<br>prozess | Klasse 8 - LB 2:<br>Daten verarbeiten<br>(verbale oder schemati-<br>sche Darstellung von Ab-<br>läufen in einer didaktisch<br>reduzierten Lernumge-<br>bung)                   |   |  |
|  | Klasse 9/10 - LB 4:<br>Algorithmen und Pro-<br>gramme<br>- algorithmische Grund-<br>strukturen: Sequenz, Se-<br>lektion, Zyklus<br>- Modularisierung<br>- Problemlöseprozesses |   |  |
|  | Grundkurs 11/12 (sprachli-<br>ches Profil) - LB 4:<br>Algorithmen und Pro-<br>gramme<br>(zur Wdh., eigentlich<br>höhere Programmier-<br>sprache wie Java oder C<br>gefordert)  | Klasse 12 - LB 2:<br>Algorithmen und Pro-<br>gramme<br>(zur Einführung/Wdh.,<br>eigentlich höhere Pro-<br>grammiersprache wie Java<br>oder C gefordert) | Grundkurs 11/12 - LB 3:<br>Algorithmen und Pro-<br>gramme<br>(zur Wdh., eigentlich<br>höhere Programmier-<br>sprache wie Java oder C<br>gefordert) |

**zusätzlich:** fächerübergreifender Unterricht (Lernspiele/Animationen/... für andere Fächer)

#### 3. Lernziele

##### Kognitive Lernziele:

Die Schüler:innen erhalten Einblick in die Modularisierung. (GY)

Die Schüler:innen kennen grundlegende Programmstrukturen (Folge, Wiederholung, Verzweigung). (OS)

Die Schüler:innen wenden ausgewählte Problemlösungsstrategien in einer höheren Programmiersprache an. (FOS)

Die Schüler:innen wenden ihre Kenntnisse zum Problemlöseprozess auf das selbstständige Lösen einfacher Probleme in einer einfachen grafischen Programmierungsumgebung an. (OS)

## **Psychomotorische Lernziele:**

Die Schüler:innen implementieren in einer Programmierumgebung die Lösung eines Problems unter Anwendung des Problemlöseprozesses. (GY)

Die Schüler:innen beherrschen die Umsetzung der algorithmischen Grundstrukturen Sequenz, Selektion und Zyklus beim Programmieren. (GY)

Die Schüler:innen beherrschen ausgewählte Problemlösestrategien in einer Programmiersprache. (BGy)

## **Affektive Lernziele:**

Die Schüler:innen bewerten ihre Resultate beim Anwenden des Problemlöseprozesses zur Lösung eines Problems in einer Programmierumgebung kritisch. (GY)

Die Schüler:innen beachten die Anwendung des Problemlöseprozesses beim Programmieren. (alle)

Die Schüler:innen werden aufmerksam auf die Modularisierung beim Programmieren. (alle)

## **4. Kompetenzentwicklung**

### **Fachkompetenz:**

Die Schüler:innen können

- die algorithmischen Grundbausteine (Sequenz, Selektion, Zyklus) benennen und in ihrer Funktionsweise beschreiben.
- arithmetische und logische Operatoren klassifizieren.
- Problemlösestrategien beschreiben.
- das Konzept der Modularisierung beschreiben.

### **Lern-/Methodenkompetenz:**

Die Schüler:innen können,

- Problemlösungen in einer Programmiersprache umsetzen. (Bildungsstandards 8-10, Lehrplan)
- Problemlöseprozesse bei der Lösung eines Problems in einer Programmierumgebung anwenden. (Lehrplan)
- Algorithmen mit den algorithmischen Grundbausteinen realisieren. (Bildungsstandards Sek I, Sek II)
- Variablen und Wertzuweisungen verwenden. (Bildungsstandards 8-10)
- arithmetische und logische Operationen beim Problemlösen verwenden. (Bildungsstandards 8-10)

### **Sozialkompetenz:**

Die Schüler:innen können

- in verschiedenen Formen der Zusammenarbeit bei der Bearbeitung einfacher informatischer Probleme kooperieren. (Bildungsstandards Sek I)
- informatische Inhalte und Vorgehensweisen auch außerhalb des Informatikunterrichts anwenden (Bildungsstandards Sek I) / interdisziplinär arbeiten (Lehrplan).
- fachgerecht über informatische Sachverhalte kommunizieren.

### **Selbstkompetenz:**

Die Schüler:innen können

- die Darstellung und Strukturierung informatischer Sachverhalte begründen. (Bildungsstandards 5-7)
- ihre Resultate beim Anwenden des Problemlöseprozesses zur Lösung eines Problems in einer Programmierumgebung kritisch bewerten. (Lehrplan)

- in Alternativen denken, Phantasie und Kreativität einsetzen und zugleich Lösungen auf ihre Machbarkeit überprüfen. (Lehrplan)
- ihre eigenen Interessen produktiv im Informatikunterricht einbringen und ihre Motivation fördern, indem sie Programme zu selbstgewählten Themen entwerfen und implementieren.

## 5. Prinzipieller Aufbau

Es gibt eine Oberfläche, die aus verschiedenen Bereichen aufgebaut ist:

1 - **Menüleiste:** In der Menüleiste kann die Sprache ausgewählt werden, ein Scratch-Projekt gespeichert oder geladen werden, unter „Bearbeiten“ ein Turbomodus eingeschaltet werden. Außerdem gibt es Zugriff auf Tutorials zum Kennenlernen der Scratch-Umgebung und die Möglichkeit, den Namen des aktuellen Projektes zu ändern.

2 - **Bühne:** Auf der Bühne läuft das Programm ab. Hier bewegen sich die Figuren und können Bühnenbilder angezeigt werden, je nachdem was programmiert wird. Die Bühne ist als Koordinatensystem aufgebaut und kann via x- und y-Werte angesteuert werden. Ein Programm wird mit der grünen Flagge und dem roten Kreis oberhalb der Bühne gestartet und gestoppt.

3 - **Figurenliste:** In diesem Bereich werden die Figuren aufgelistet, die im Programm vorkommen. Über das Symbol unten rechts können weitere Figuren ausgewählt oder als Bilder selbst erstellt/hochgeladen werden. Oberhalb der Auflistung kann man die Anzeige jeder Figur auf der Bühne konfigurieren.

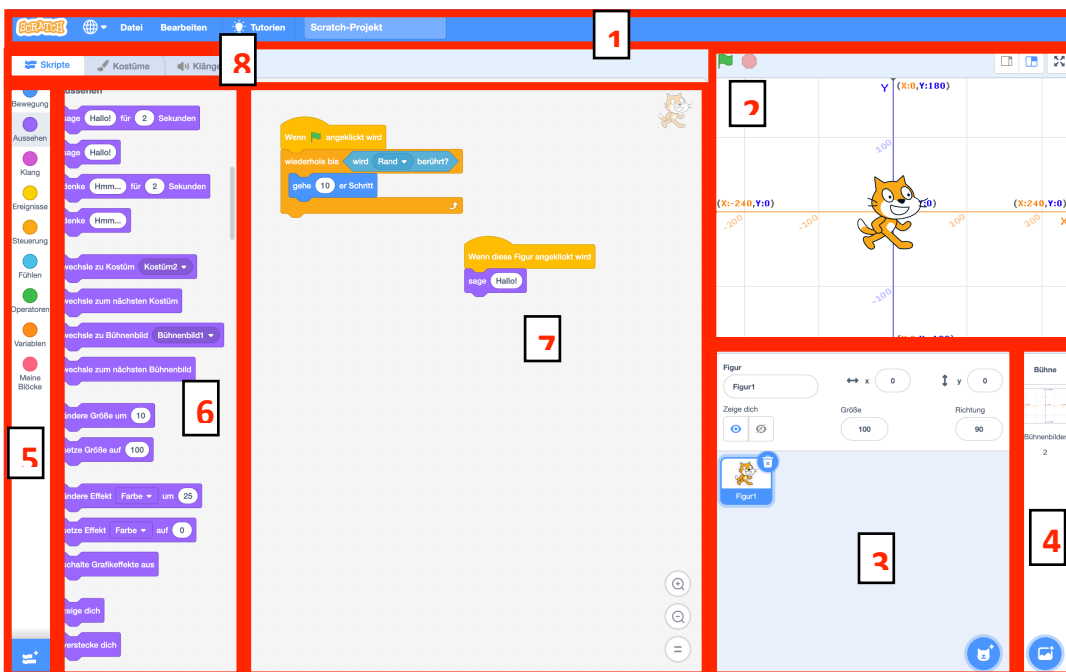
4 - **Bühnenliste:** In diesem Bereich werden die Bühnenbilder aufgelistet, die im Programm vorkommen. Über das Symbol unten können weitere Bühnenbilder ausgewählt oder als Bilder selbst erstellt/hochgeladen werden.

5 - **Blockkategorie-Bereich:** In diesem Menü kann man zwischen Blockkategorien wechseln. Es können mit dem Symbol unten weitere vorgefertigte Blockkategorien wie zu LEGO Mindstorms oder Videoerfassung hinzugefügt werden. Unter „Meine Blöcke“ können zudem eigene Blöcke erstellt werden.

6 - **Blockpalette:** Je nach Blockkategorie gibt es eine eigene Blockpalette, die farblich der Kategorie angepasst ist. Blöcke, die den Start eines Programmablaufs bewirken, haben links oben eine Auswölbung nach oben. Punkte im Programmablauf sind durch die puzzle-artige Auswölbung nach unten hin gekennzeichnet. Variablen und Werte haben abgerundete Ecken und Bedingungsabfragen sind als Sechsecke dargestellt.

7 - **Skript-Bereich:** Im Skriptbereich werden die Blöcke per Drag-&Drop hineingezogen und der Programmablauf zusammengestellt. Oben rechts wird angezeigt, zu welcher Figur/Bühne der aktuelle Programmcode ist.

8 - **Medien-Bereich:** Im Medienbereich können Kostüme der Figuren und Klänge geändert oder erstellt werden.



## 6. Handhabung

### Benennen des Projektes und Speichern:

- in der Menüleiste den Standard-Titel „Scratch-Projekt“ umbenennen
- in der Menüleiste „Datei“ anklicken → „Auf deinem Computer speichern“

### Projekt laden:

- in der Menüleiste „Datei“ anklicken → „Hochladen von deinem Computer“

### Figuren hinzufügen:

- die Maus in der Figurenliste unten rechts auf den blauen Kreis bewegen
- Bild der Figur aus Liste auswählen (Lupe), malen (Pinsel) oder als Datei einfügen



### Figur bearbeiten:

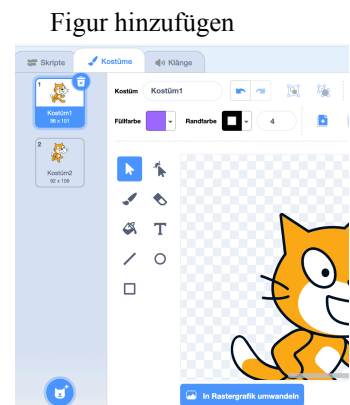
1. Figur anklicken
2. im Medienbereich unter „Kostüme“ oder „Klänge“ manuell bearbeiten, hochladen oder Ton/Bild per Webcam/Mikrofon aufnehmen

### Figuren entfernen: Papierkorb-Symbol neben Figur in Figurenliste klicken

### Figur positionieren:

1. Figur anklicken
2. über der Figurenliste x- und y-Position ändern, als auch Größe, Ausrichtung und Sichtbarkeit

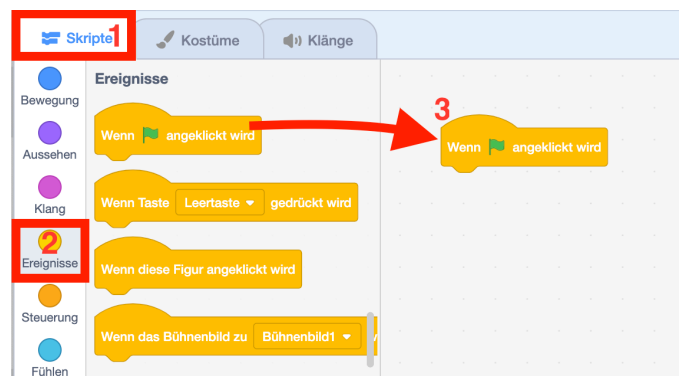
### Bühne ändern/hinzufügen: s. Figuren hinzufügen/bearbeiten/entfernen, aber in der Bühnenliste und im Medienbereich mit Bühnenbildern statt Kostümen



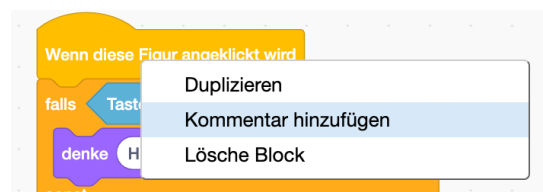
### Programmieren:

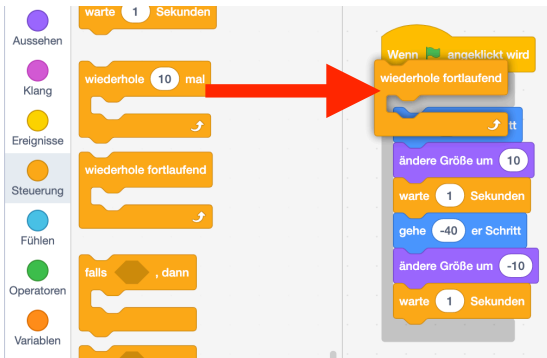
1. Element (Figur/Bühne) anklicken, zu dem ein Programmablauf hinzugefügt werden soll, dann auf „Skripte“ oben links klicken. (1 im Bild rechts)
2. Blockkategorie auswählen (2) oder im Blockbereich scrollen.
3. per Drag & Drop Elemente aus der Blockpalette in den Skriptbereich ablegen (3)
  - Programme starten mit mind. einem Ereignisblock → mehrere Blöcke = parallele Programmausführung
  - zwischen einzelnen Figuren/Bühnen kann durch Nachrichten kommuniziert werden (unter „Ereignisse“) - diese reagieren mit dem Ereignisblock „Wenn ich <nachricht\_xy> empfangen“ (s. nächste Seite)
  - Kommentieren einzelner Blöcke durch Klicken der rechten Maustaste auf dem Block und „Kommentar hinzufügen“ (s. rechts)
  - Löschen von Blöcken durch Ziehen in Blockpalette oder Rechtsklick und „Lösche Block“
  - Zyklen und Selektion sind unter „Steuerung“ → müssen über die jeweilige Sequenz gezogen werden → zugehörige Sequenzen werden geschachtelt dargestellt (s. nächste Seite)

Kostüm bearbeiten



Blöcke per Drag & Drop ziehen





Zyklus/Selektion einfügen



Nachrichten senden und empfangen

### Variablen:

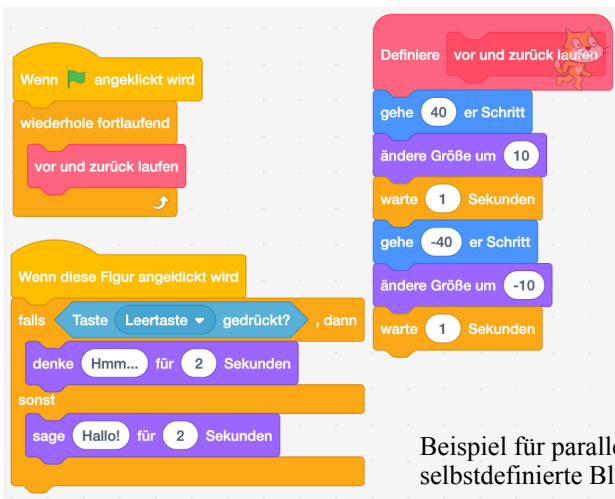
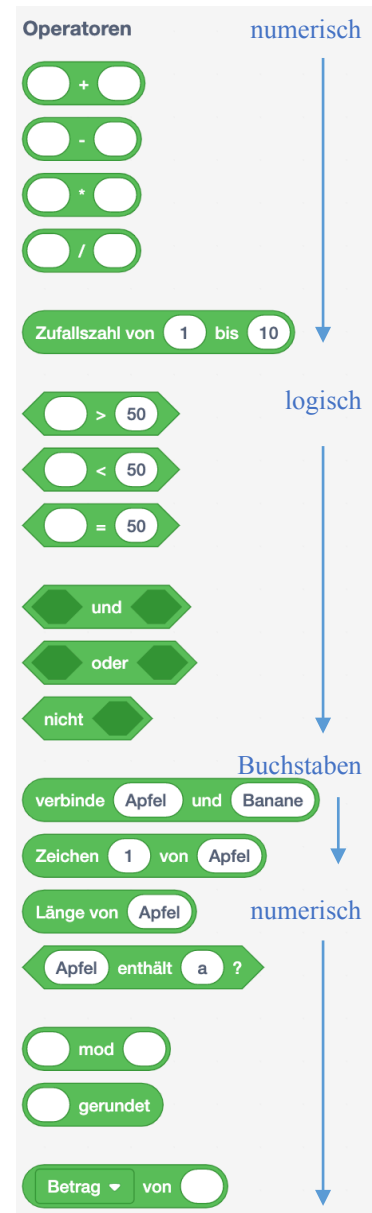
- es existieren vorgefertigte Variablen (unter „Fühlen“: x-,y-Position, Richtung, Lautstärke, ...)
- eigene Variablen werden unter „Variablen“ im Blockkategorie-Bereich erstellt und sind numerisch oder Zeichenketten
- eigene Variablen werden auf der Bühne angezeigt und können z.B. zum Zählen von Punkten in einem Spiel oder zum Zeitmessen genutzt werden.

### Operatoren

- logische Ausdrücke sind als Sechsecke gestaltet und in sechseckige Lücken einzusetzen
- numerische Ausdrücke sind als abgerundete Vierecke bzw. Kreise gestaltet und in die runden Lücken einzusetzen

**eigene Blöcke erstellen:** im Sinne der Modularisierung, um häufige gemeinsame Programmierbausteine zusammenzufassen oder das Projekt übersichtlicher zu gestalten

- „Meine Blöcke“ im Blockkategorie-Bereich auswählen
- auf „Neuer Block“ klicken
- im Pop-Up-Fenster Elemente zum Block hinzufügen und Blockname ändern
- im Skriptbereich Blockdefinition wie gewohnt durch Drag&Drop programmieren




Beispiel für parallele Programmausführung und selbstdefinierte Blöcke (rot)

### Starten/Stoppen eines Programms:

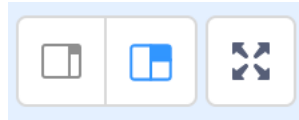
- Beginn mit Klicken der grünen Flagge über der Bühne, wenn dieses Startereignis gewählt wurde
- Ende mit Klicken des roten Sechsecks über der Bühne

### Erweiterungen hinzufügen (Musikinstrumente, Malstift, Videoerfassung, LEGO, GoogleTranslate, ...):

- im Blockkategorie-Bereich ganz unten auf  klicken
- Hinweis: LEGO Mindstorms funktioniert mit Roberta besser, da dort alle Sensoren genau abgestimmt sind

### Darstellung ändern:

- mit den Symbolen oben rechts über der Bühne
- standardmäßig ist das mittlere eingestellt
- Skriptbereich vergrößern → linkes Symbol
- Bühne im Vollbild → rechtes Symbol



### Weitere Hinweise:

- unter <https://scratch.mit.edu/> können Schüler:innen die Projekte anderer anschauen und auch selber Konten erstellen und ihre Projekte hochladen
- es gibt die Möglichkeit von Lehrerkonten: <https://scratch.mit.edu/educators#teacher-accounts>, um Schülerkonten zu erstellen, sowie Projekte und Kommentare zu verwalten
- Scratch benötigt keinen Download, da es auch einen Online-Editor (ohne Login) hat: <https://scratch.mit.edu/projects/editor/>
- durch eingebaute Tutorien (Online oder in der Download-Version) kann man Scratch nach dem Flipped-Classroom-Konzept einführen → über ein Tutorial allein den Umgang kennenlernen, im Unterricht an einem Projekt arbeiten

## 7. Screencast

Link und QR-Code zum Manual

## 8. Aufgaben

### Aufgabe 1: Kennenlernen der Programmierumgebung

**10 min - Ziele:** Du lernst die Oberfläche intensiver kennen und findest heraus, wo du welche Elemente finden kannst. Du analysierst einen beispielhaften Programmablauf mit Variable, Schleife und Verzweigung.

1. Füge das Objekt „Jellyfish“ der Bühne hinzu. Ändere die Größe auf 50.
2. Entferne die Katzen-Figur.
3. Wähle als Bühnenbild „Underwater 2“. Deine Bühne sollte dann wie Bild 1 aussehen.
4. Klicke auf den Jellyfish in der Figurenübersicht. Füge dann den Programmablauf aus Bild 2 zum „Jellyfish“ hinzu.



5. Starte das Programm durch Klicken der grünen Flagge und klicke mehrmals die Leertaste. Beschreibe in wenigen Worten, was das Programm macht:
  - Benenne, wo sich in dem Programm ein Zyklus befindet und welche Funktion er hat.
  - Benenne, wo sich in dem Programm eine Selektion befindet.
6. Speichere das Programm unter „Wasserwelt.sb3“ auf deinem Computer.

### Aufgabe 2: eigenständige Erweiterungen

**30 min - Ziele:** Du wendest die Prinzipien Folge, Schleife und Verzweigung selbstständig an, um einen vorgegebenen Programmablauf zu erreichen. Du kannst eine Variable nutzen.

1. Füge folgende Bewegungen hinzu:
  - a) Ab Spielbeginn soll sich der Jellyfish fortlaufend in 10er Schritten weiter bewegen.
  - b) Wenn er den Bühnenrand erreicht, soll er vom Rand abprallen und dabei seine Richtung ändern.
2. Füge einen Kostümwechsel hinzu:  
Jetzt soll der Jellyfish die Farbe nicht mehr wechseln, wenn die Leertaste gedrückt wird, sondern wenn er den Bühnenrand erreicht.
3. Erstelle eine Variable:
  - a) Erstelle eine neue Variable namens „zähler“.
  - b) Setze die Variable zu Beginn des Spiels auf 0.
  - c) Erhöhe die Variable um 1, wenn der Rand berührt wird.
4. Füge ein Spielende hinzu:
  - a) Erstelle beim Jellyfish einen neuen Block, der mit dem Klicken der Fahne startet.
  - b) Wenn 10 Sekunden vorbei sind, soll der Jellyfish sich verstecken und alles gestoppt werden.

- c) Füge am Anfang des Programmes die Anweisung „zeige dich“ hinzu, damit der Jellyfish zum Programmbeginn immer sichtbar ist.

### Zusatzaufgaben

**10 min - Ziel:** *Du vertiefst deine Fähigkeiten in Bezug auf Schleifen und Bedingungen/Verzweigungen.*

**EINFACH:** Spielablauf ändern

Gehe zu dem Block, bei dem die Zeit abgezählt und das Spiel beendet wird.

Ändere den Block, sodass der Jellyfish 6 Mal 10 Sekunden wartet und jeweils nach 10 Sekunden der Klang „ocean wave“ abgespielt wird. Nach dem 6. Mal soll der Jellyfish sich verstecken und alles gestoppt werden.

**MITTEL:** Countdown hinzufügen

Füge eine Variable countdown hinzu, die als Startwert die Anzahl der Sekunden hat, wie lange das Programm dauern soll und die mit jeder Sekunde 1 weniger wird. Lasse sie auf der Bühne anzeigen. Das Spiel soll jetzt zu Ende sein, sobald der countdown bei 0 ist.

**SCHWER:** Spiel entwickeln

Entwickle das Programm zu einem Spiel weiter, bei dem der Benutzer oder eine andere Figur den Jellyfish so oft wie möglich fangen muss - oder setze eine eigene Idee um!

## 9. Lösungen

### Aufgabe 1

5.a. = „wiederhole fortlaufend“, um ununterbrochen zu warten, ob die Leertaste gedrückt wird

5.b = „falls Taste „Leertaste“ gedrückt, dann → nur dann findet Kostümwechsel statt

### Aufgabe 2

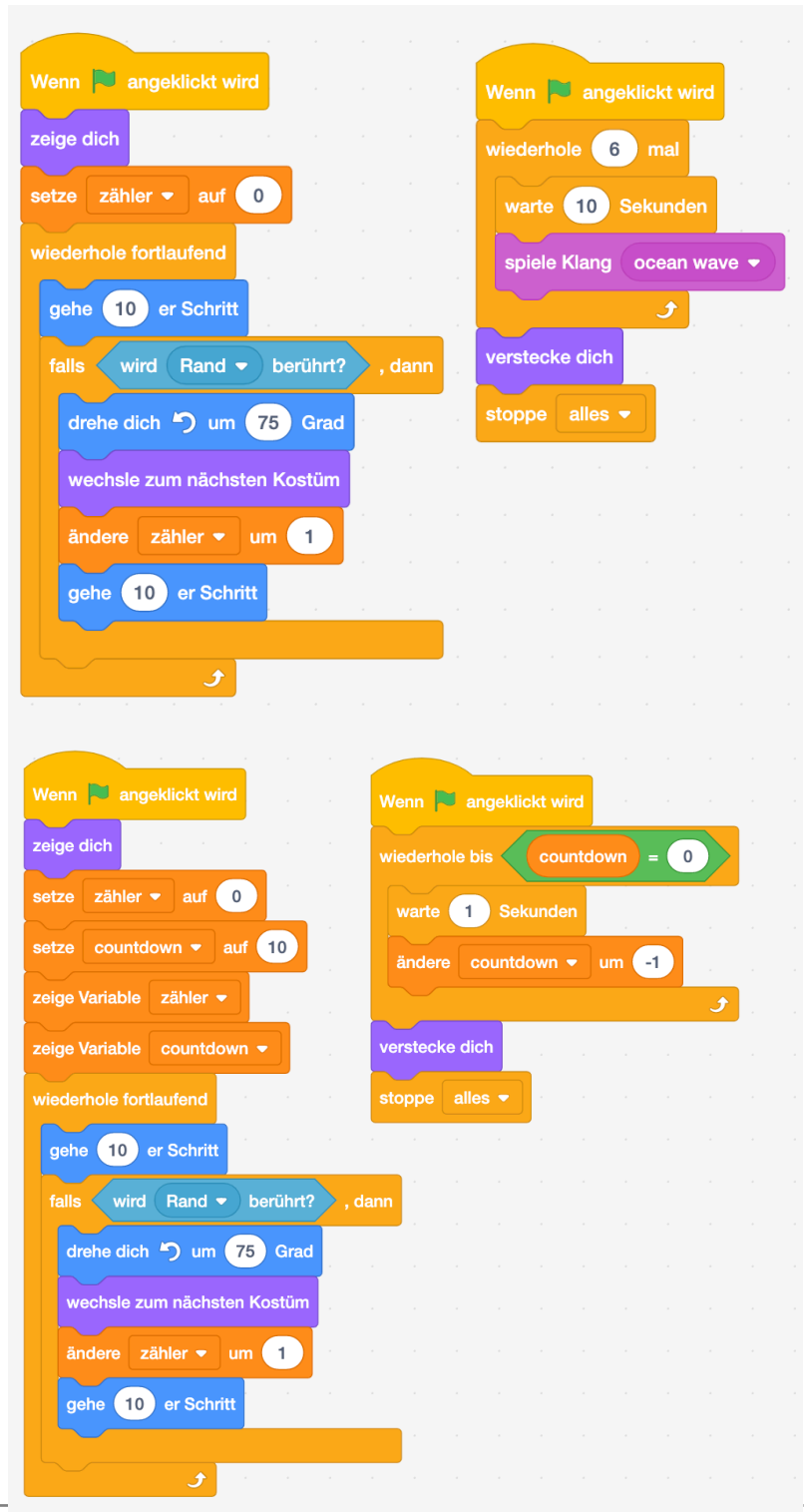
### Zusatz - einfach



```
Wenn grüner Flagge angeklickt wird
  setze zähler auf 0
  wiederhole fortlaufend
    gehe 10 er Schritt
    falls wird Rand berührt?, dann
      drehe dich um 75 Grad
      wechsele zum nächsten Kostüm
      ändere zähler um 1
      gehe 10 er Schritt
  stoppe alles
```

```
Wenn grüner Flagge angeklickt wird
  warte 10 Sekunden
  verstecke dich
  stoppe alles
```



```
Wenn grüner Flagge angeklickt wird
  zeige dich
  setze zähler auf 0
  wiederhole fortlaufend
    gehe 10 er Schritt
    falls wird Rand berührt?, dann
      drehe dich um 75 Grad
      wechsele zum nächsten Kostüm
      ändere zähler um 1
      gehe 10 er Schritt
  stoppe alles
```

```
Wenn grüner Flagge angeklickt wird
  wiederhole 6 mal
    warte 10 Sekunden
    spiele Klang ocean wave
  verstecke dich
  stoppe alles
```

```
Wenn grüner Flagge angeklickt wird
  zeige dich
  setze zähler auf 0
  setze countdown auf 10
  zeige Variable zähler
  zeige Variable countdown
  wiederhole bis countdown = 0
    warte 1 Sekunden
    ändere countdown um -1
  verstecke dich
  stoppe alles
```

```
Wenn grüner Flagge angeklickt wird
  gehe 10 er Schritt
  falls wird Rand berührt?, dann
    drehe dich um 75 Grad
    wechsele zum nächsten Kostüm
    ändere zähler um 1
    gehe 10 er Schritt
  stoppe alles
```

### Zusatz - mittel