

Block 1: Ankommen & Scratch einrichten (20 min)

Teilnehmer-Handout. Kästen mit  sind Hinweise für die Workshop-Leitung und werden vor dem Druck entfernt. Platzhalter für Bilder:  [Bild: ...]

Was du in diesem Block lernst

- Was Programmieren ist und was eine Programmiersprache ist
- Der Scratch-Klasse per QR-Code beitreten und dein eigenes Spielprojekt holen
- Die Scratch-Oberfläche kennen: Bühne, Figuren, Blöcke, Skripte
- Dein erstes Skript bauen: Das Boot sagt „Hallo!“
- Dein Projekt speichern

1.1 Was ist Programmieren? (3 min)

Stell dir einen Helfer vor, der **alles** für dich macht, aber **nichts** von selbst versteht. Sagst du „Mach mir ein Brot“, steht er nur da. Du musst sagen: Nimm das Brot. Nimm das Messer. Streich die Butter auf die obere Seite. Genau in dieser Reihenfolge. Vergisst du einen Schritt oder sagst du ihn falsch herum, kommt Unsinn heraus. Und der Helfer macht den Unsinn trotzdem, ohne zu meckern.

Ein Computer ist so ein Helfer. **Programmieren** heißt: ihm Schritt für Schritt aufschreiben, was er tun soll. So eine Liste von Schritten ist ein **Programm**. Der Computer ist dabei sehr schnell und sehr gehorsam, aber er rät nie, was du gemeint hast. Wenn dein Spiel heute etwas Komisches macht, liegt es fast immer daran: Das Programm sagt genau das, du hast es nur anders gemeint. Das ist normal. Programmieren ist zur Hälfte Aufschreiben und zur anderen Hälfte Nachschauen, warum der Helfer etwas anderes gemacht hat.

Programmiersprachen

Der Helfer versteht kein Deutsch. Er versteht nur ganz bestimmte Wörter in ganz bestimmter Form: eine **Programmiersprache**. Davon gibt es viele, so wie es Deutsch, Englisch und Türkisch gibt: Python, JavaScript, Java, C und noch mehr. Alle können im Kern dasselbe sagen („mach das“, „wiederhole das“, „falls das, dann das“), nur mit anderen Wörtern. Wer eine kann, lernt die nächste viel schneller.

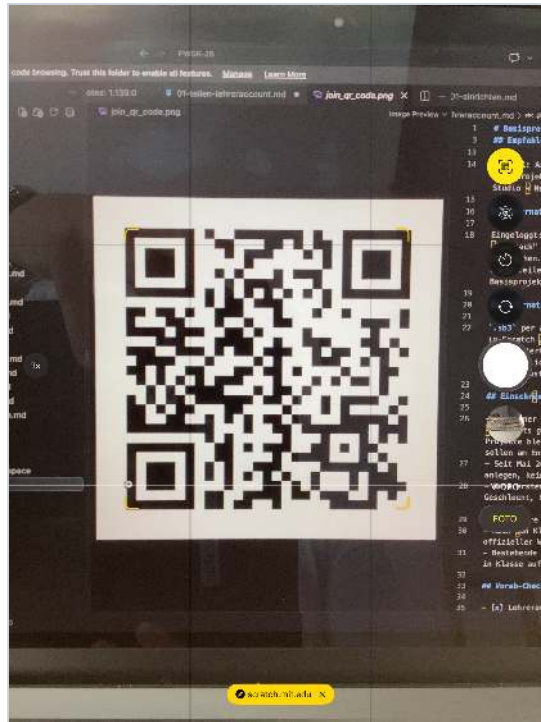
Scratch ist eine Programmiersprache, bei der die Wörter als **Blöcke** vor dir liegen. Du tippst nichts, du steckst zusammen, und die Blöcke passen nur dort, wo sie Sinn ergeben. So kannst du dich nicht vertippen und siehst sofort, was zusammengehört. Was du heute in Scratch lernst, nämlich Schleifen, Bedingungen, Variablen und Nachrichten, steckt genauso in jedem Handyspiel und jeder App. Nur in anderer Schrift.

 Leitung: Das Helfer-Bild trägt den ganzen Tag: „Was steht im Programm?“ statt „Was hast du gemeint?“. Wenn zwei Minuten übrig sind, die Roboter-Demo: Du bist der Helfer, die Gruppe sagt dir, wie man ein Marmeladenbrot macht. Du führst wörtlich aus (Messer ins geschlossene Glas, Brot mit der Tüte auf den Teller, Marmelade auf die Unterseite). Lacher garantiert, und der Satz „Der Computer macht genau, was dasteht“ sitzt. Pantomime reicht, Requisiten sind nicht nötig. Danach eine Frage in die Runde: „Welche Programmiersprachen kennt ihr schon vom Hören?“ Antworten einsammeln, nicht erklären.

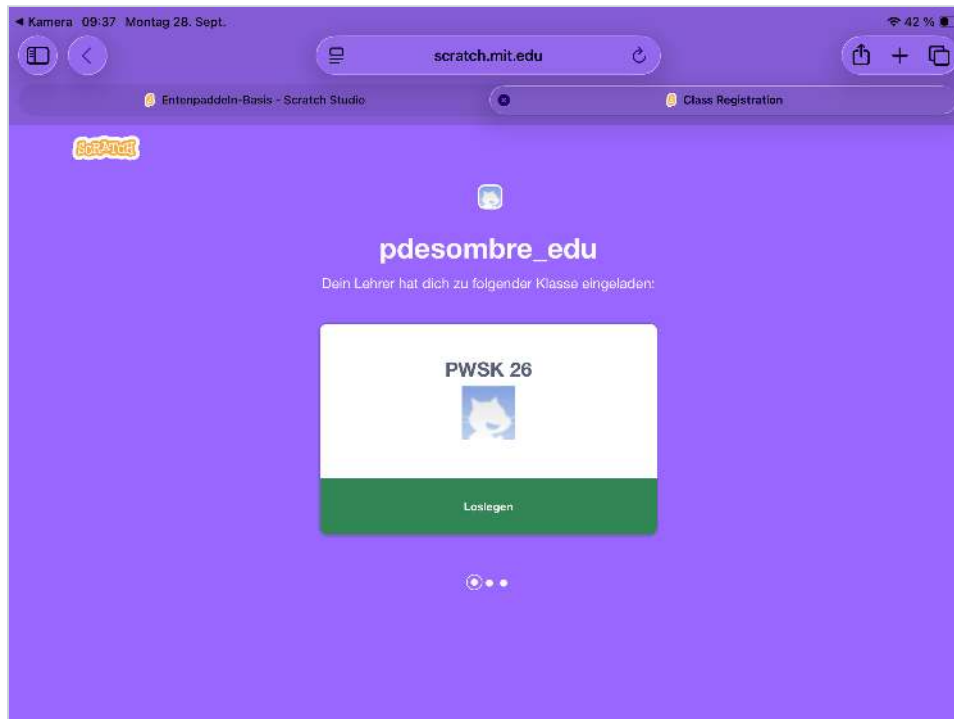
1.2 Der Klasse beitreten (5 min)

Du legst dir gleich deinen eigenen Scratch-Account an. Der QR-Code am Beamer bringt dich direkt zur richtigen Seite — du landest automatisch in unserer Klasse.

1. Halte das iPad **quer** (Landschaft). Scratch braucht Platz.
2. Öffne die **Kamera-App** und halte sie auf den **QR-Code am Beamer**.
3. **Unten** erscheint ein gelbes Feld mit **scratch.mit.edu**. **Tippe darauf** — Safari öffnet die Anmeldeseite.



4. Du siehst die Einladung zur Klasse **PWSK 26**. Tippe auf den grünen Knopf „**Loslegen**“.



5. **Denk dir einen Benutzernamen aus.** 3–20 Zeichen, nur Buchstaben, Zahlen, - und _. Fang mit **ente** an, dann bleibt unsere Klasse zusammen, zum Beispiel ente_bltitz7. Nimm **nicht** deinen echten Namen — Scratch weist im blauen Kästchen selbst darauf hin.
6. **Denk dir ein Passwort aus.** Mindestens 6 Zeichen. Setz den Haken bei „**Passwort anzeigen**“, dann siehst du, was du tippst.
7. **Schreib beides sofort auf deine Karte.** Ohne Passwort kommst du nach einer Pause nicht zurück in dein Projekt.

Kamera 09:37 Montag 28. Sept. scratch.mit.edu 42 %

Benutzerkonto

Mit Scratch kannst du Spiele, Animationen und Geschichten erschaffen. Ein Benutzerkonto zu erstellen ist einfach und kostenlos. Fülle dazu das folgende Formular aus. Hast du bereits ein Scratch-Benutzerkonto? [i](#)

Wähle einen Benutzernamen
Vielleicht dein Lieblingsessen, dein liebstes Hobby oder Tier und ein paar Zahlen?

Passwort

☐ Passwort anzeigen

Nächster Schritt

Benutze nicht deinen echten Namen

8. „**Nächster Schritt**“ tippen. Es folgen noch ein paar Fragen: **Geburtsmonat und -jahr, Geschlecht, Land** (Deutschland). Ausfüllen, abschicken — fertig. Du bist angemeldet.

Klappt der QR-Code nicht? Auf deiner Karte steht der Link auch zum Abtippen.

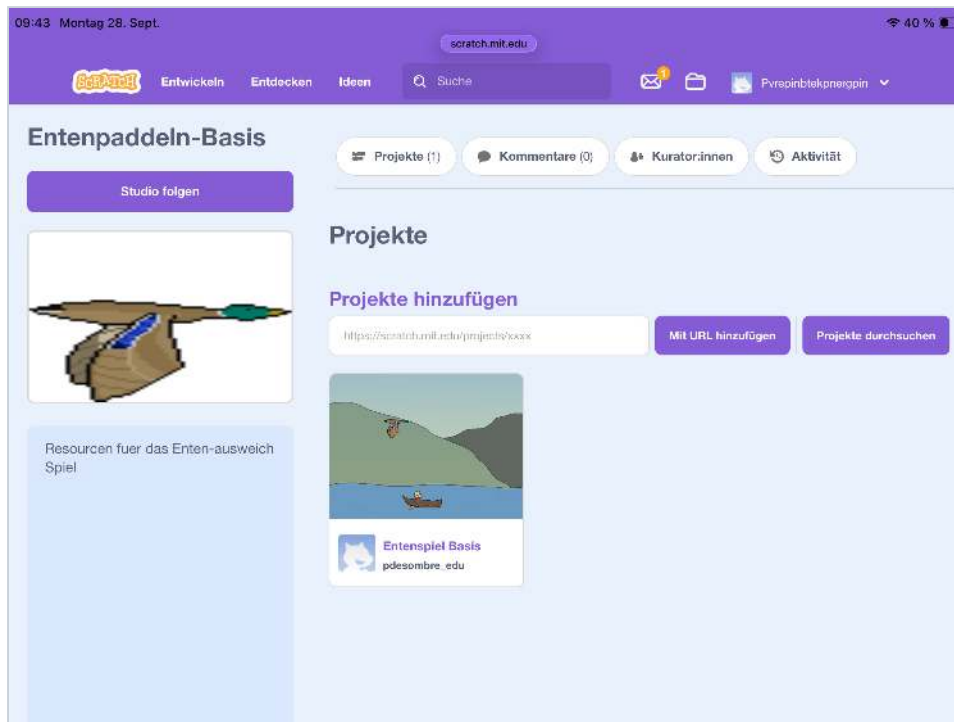


🎓 **Leitung:** Anmeldelink vorab in der Klasse erzeugen (scratch.mit.edu → Meine Klassen → Klasse → Tab „Schüler“ → „generate“), als QR-Code auf eine Folie legen und **während des ganzen Blocks stehen lassen** — Nachzügler kommen sonst nicht mehr dran. Den Link zusätzlich auf die Karte drucken (QR-Scan scheitert bei schlechter Beamer-Ausleuchtung oder deaktiviertem Code-Scannen). Karten: pro Platz eine Karte mit Link, zwei Leerzeilen für Nutzernamen und Passwort, und Platz für den Vornamen — die Karten am Ende einsammeln oder abfotografieren, sonst weißt du später nicht, welcher Account zu wem gehört. Der Nutzernamen muss Scratchweit eindeutig sein, „ente1“ ist längst vergeben — deshalb der Vorschlag mit Zusatz. Passwort vergessen: du kannst es im Lehreraccount zurücksetzen. Halte 3–4 Reserve-Accounts bereit (vorab per CSV), falls jemand die Anmeldung gar nicht schafft. Nach dem Workshop den Link nicht weiter verteilen; bei Bedarf über „generate“ einen neuen erzeugen. WLAN-Zugang vorher auf jedem iPad testen. Safari-Zoom: falls die Seite riesig/klein ist, mit zwei Fingern zurückzoomen oder Seite neu laden. Orga-Betreuer: geht während der Anmeldung von Platz zu Platz, Fach-Betreuer bleibt am Beamer.

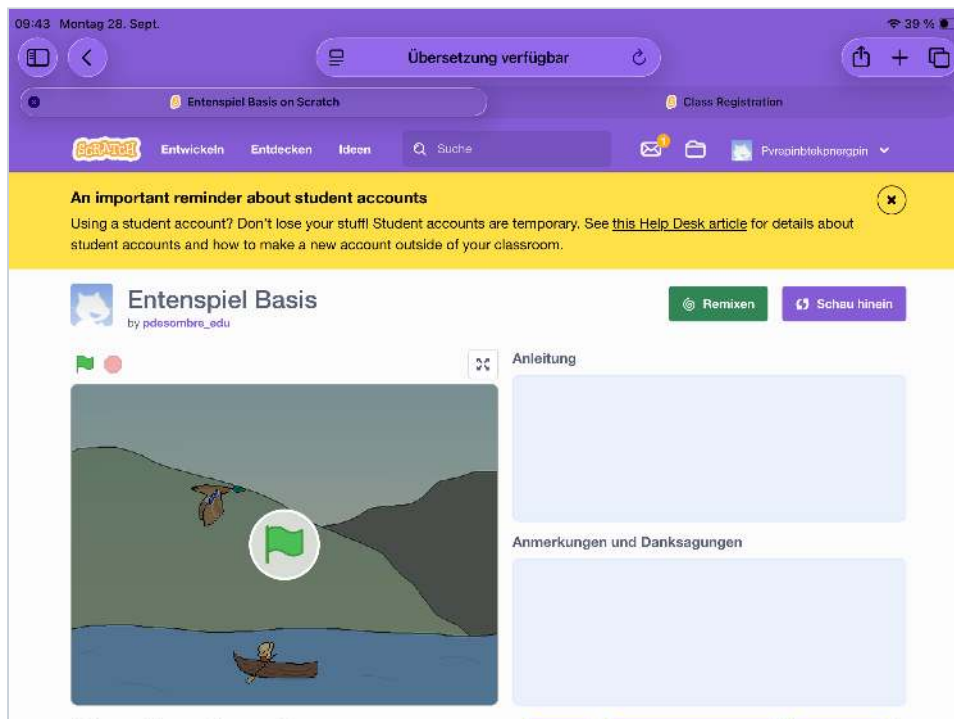
1.3 Dein Spielprojekt holen (4 min)

Das Grundgerüst des Spiels liegt schon bereit — mit Boot, Ente, Kacke, Hintergrund und allen Geräuschen. Du machst dir eine eigene Kopie davon. Das heißt bei Scratch **„Remix“**.

1. Am Beamer erscheint jetzt der **zweite QR-Code**. Scanne ihn wie vorhin mit der Kamera-App. Er öffnet das Studio **„Entenpaddeln-Basis“**. (Der Link steht auch auf deiner Karte.)
2. Tippe im Studio auf das Projekt **„Entenspiel Basis“**.



3. Tippe auf den grünen Knopf **„Remixen“**. Jetzt gehört die Kopie dir.



4. Ändere den Namen oben links in **Entenspiel <dein Benutzername>**, zum Beispiel Entenspiel ente_blitz7.
5. Tippe auf **„Jetzt speichern“**.

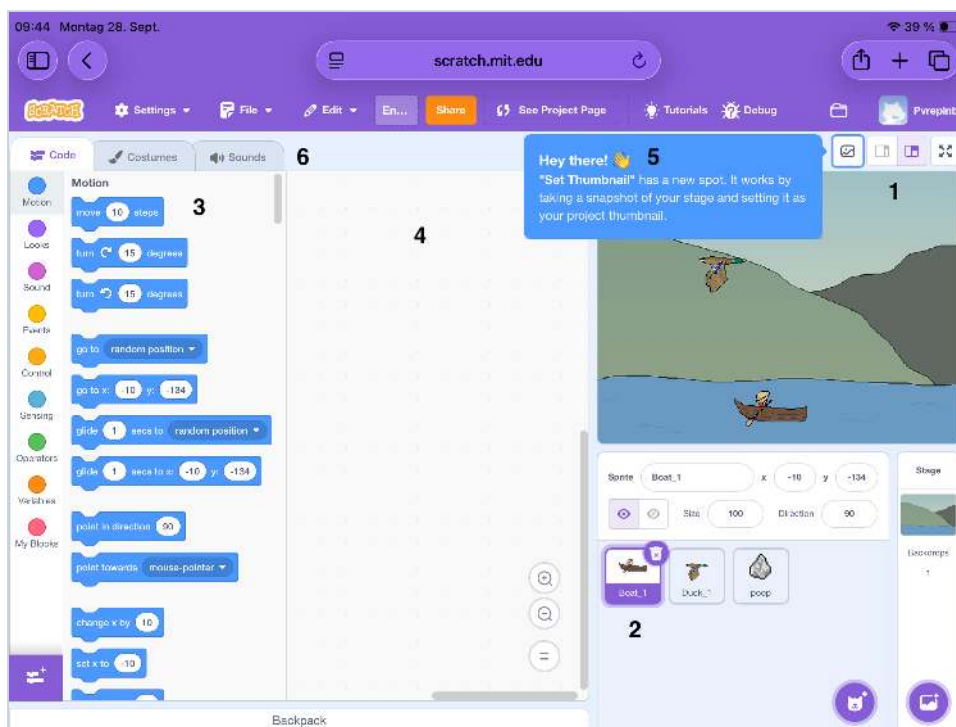
Der gelbe Balken „An important reminder about student accounts“ darf weg — tippe auf das **×**. Er erinnert nur daran, dass dein Workshop-Konto vorübergehend ist.

Siehst du den Knopf „Remixen“ nicht? Dann bist du nicht angemeldet. Oben rechts prüfen, ob dein Nutzernamenasteht.

Leitung: Am Beamer einmal komplett vormachen, danach den Studio-QR stehen lassen. Zweiter Weg, falls der QR nicht geht: Nutzernamenasteht oben rechts → „Meine Klasse“ → Tab „Studios“ → „Entenpaddeln-Basis“. Das Startprojekt muss **geteilt** sein, sonst fehlt der Knopf „Remixen“ — das ist der häufigste Fehler in diesem Schritt. Der Remix heißt automatisch „Entenspiel Basis remix“; Umbenennen ist wichtig, damit du die Projekte später im Studio auseinanderhalten kannst. Die Teilnehmer legen ihr eigenes Projekt am Ende (Block 10) ebenfalls ins Studio — dann siehst du alle Stände an einer Stelle.

1.4 Die Scratch-Oberfläche (5 min)

Scratch besteht aus vier Bereichen. Finde sie auf deinem Bildschirm:



Nr.	Bereich	Was es ist
1	Bühne (rechts oben)	Hier läuft dein Spiel. Was du programmierst, siehst du hier.
2	Figurenliste (rechts unten)	Alle Figuren im Spiel: Boot, Ente, Kacke. Die Figur, die du antippst, programmierst du gerade. Daneben die Bühne selbst (Hintergrund).
3	Blockpalette (links)	Alle Befehle, sortiert nach Farben. Bewegung = blau, Aussehen = lila, Klang = pink, Ereignisse = gelb, Steuerung = orange, Fühlen = hellblau, Operatoren = grün, Variablen = dunkelorange.
4	Skriptbereich (Mitte)	Hier baust du deine Programme aus Blöcken zusammen, wie Lego-Steine.
5	Grüne Flagge / Stopp (über der Bühne)	Grüne Flagge startet das Spiel, rotes Stoppschild beendet es.
6	Tab „Skripte“, „Kostüme“, „Klänge“ (links oben)	Skripte = Programm. Kostüme = Aussehen der Figur. Klänge = Geräusche der Figur.

Wichtige Wörter

- Figur** (englisch: Sprite): Alles, was sich auf der Bühne bewegen kann. Bei uns: Boot, Ente, Kacke.
- Block**: Ein einzelner Befehl. Blöcke haben Farben, die zur Kategorie passen.
- Skript**: Mehrere Blöcke, die zusammenhängen. Ein Skript beginnt fast immer mit einem gelben **Hut-Block** (rund oben), zum Beispiel „Wenn angeklickt wird“.

Mini-Aufgabe: Erkunden (2 min)

- Tippe in der Figurenliste auf **Ente**. Was ändert sich im Skriptbereich? Und in der Blockpalette oben?

2. Tippe auf den Tab „**Kostüme**“. Wie viele Kostüme hat die Ente?
3. Tippe zurück auf „**Skripte**“ und dann in der Figurenliste wieder auf **Boot**.

1.5 Bedienung mit dem Finger (2 min)

Scratch ist eigentlich für die Maus gebaut. Auf dem iPad gilt:

Du willst ...	So geht's
Block in den Skriptbereich holen	Block in der Palette antippen, halten, ziehen
Block an einen anderen anhängen	Ziehen, bis ein grauer Schatten erscheint, dann loslassen
Block wieder loswerden	Zurück in die Blockpalette ziehen
Block kopieren oder löschen	Block lange gedrückt halten → Menü „Duplizieren“ / „Block löschen“
Zahl oder Text ändern	Auf das weiße Feld tippen, Tastatur erscheint
Skriptbereich vergrößern/verkleinern	Die Knöpfe + - = unten rechts im Skriptbereich. Nicht mit zwei Fingern zoomen, das zoomt die ganze Webseite.
Bühne größer sehen	Vollbild-Knopf rechts über der Bühne. Zurück mit dem gleichen Knopf.

🎓 **Leitung:** Häufigster Fehler in den ersten Minuten: Zwei-Finger-Zoom von Safari. Zeigen, wie man das zurücksetzt. Zweiter Fehler: Block landet nicht „eingerastet“ und liegt frei daneben. Auf den Schatten hinweisen. **Beamer läuft über Laptop, nicht iPad.** Beim Vormachen jedes Mal übersetzen: Rechtsklick = lange drücken, Maus ziehen = Finger ziehen, Tastatur tippen = Feld antippen, Tastatur erscheint. Diese Tabelle am besten ausgedruckt neben jedes iPad legen. Ideal: ein iPad in der Hand haben und die Geste parallel zeigen.

1.6 Übung: Das Boot sagt „Hallo!“ (4 min)

Jetzt dein erstes Skript. Wähle in der Figurenliste das **Boot**.

Schritt 1: Den Start-Block holen

1. Tippe links auf die Kategorie **Ereignisse** (gelb).
2. Ziehe den Block „**Wenn**  **angeklickt wird**“ in den Skriptbereich.

Schritt 2: Den Sprech-Block anhängen

1. Tippe auf die Kategorie **Aussehen** (lila).
2. Ziehe den Block „**sage (Hallo!) für (2) Sekunden**“ unter den gelben Block, bis er einrastet.

Dein Skript sieht jetzt so aus:

```
Wenn  angeklickt wird
  sage (Hallo!) für (2) Sekunden
```

Schritt 3: Testen

Tippe auf die **grüne Flagge**. Das Boot bekommt eine Sprechblase mit „Hallo!“ für 2 Sekunden.

Schritt 4: Verändern

1. Tippe auf das Feld „**Hallo!**“ und schreib etwas anderes, zum Beispiel Achtung, Enten!
2. Ändere die **2** auf **5**. Was wird passieren?
3. Grüne Flagge, testen.

Was ist passiert?

Das Skript ist ein Befehl an das Boot:

„**Wenn** jemand die grüne Flagge drückt, **dann** zeige 2 Sekunden lang eine Sprechblase.“

- Der gelbe Hut-Block ist ein **Ereignis**. Er wartet, bis etwas passiert (Flagge gedrückt), und startet dann alles, was unter ihm hängt.
- Der lila Block ist ein **Befehl**. Er wird ausgeführt, wenn er an der Reihe ist.

- Blöcke ohne Hut-Block darüber passieren bei der grünen Flagge **nicht**. Probiere es: Zieh den lila Block vom gelben ab, drücke die Flagge. Nichts. Tippe direkt auf den lila Block: Er läuft trotzdem einmal. Das ist praktisch zum Ausprobieren.
- Was sind Ereignisse und Aktionen in deinem Alltag?

1.7 Speichern (1 min)

Scratch speichert **nicht** automatisch bei jedem Schritt. Gewöhn dir an:

Nach jeder Übung: „Jetzt speichern“ tippen.

Oben in der Menüleiste steht danach kurz „Projekt gespeichert“.



[Bild: Menüleiste mit „Jetzt speichern“ markiert]

 **Leitung:** Bei Seite-neu-laden oder Safari-Absturz ist ungespeicherte Arbeit weg. Vor jeder Pause gemeinsam speichern lassen.

Herausforderung (wenn du schneller bist)

Such dir eine aus:

1. **Gespräch:** Hänge einen zweiten „sage ... für ... Sekunden“-Block an. Das Boot sagt zwei Sätze nacheinander.
2. **Geräusch:** Kategorie **Klang** (pink) → Block **„spiele Klang (...) ganz“** anhängen. Welche Klänge hat das Boot schon?
3. **Antippen:** Kategorie **Ereignisse** → Block **„Wenn diese Figur angeklickt wird“** in den Skriptbereich, darunter ein „sage“-Block. Tippe dann auf das Boot auf der Bühne.
4. **Die Ente redet:** Wähle die **Ente** in der Figurenliste und bau ihr ein eigenes Skript. Merke: Jede Figur hat ihre eigenen Skripte.

Checkliste Block 1

- ☐ Ich bin bei Scratch angemeldet.
- ☐ **Nutzername und Passwort stehen auf meiner Karte.**
- ☐ Ich habe mein eigenes Projekt Entenspiel <Benutzername> (Remix).
- ☐ Ich weiß, wo Bühne, Figurenliste, Blockpalette und Skriptbereich sind.
- ☐ Das Boot sagt etwas, wenn ich die grüne Flagge drücke.
- ☐ Ich habe auf „Jetzt speichern“ getippt.

Weiter geht's: In Block 2 bringen wir das Boot dazu, deinem Finger zu folgen.

🎓 Leitung – Ablauf Block 1 (20 min)

Min	Was
0–2	Begrüßung, Ziel des Tages zeigen: fertiges Spiel am Beamer 1 min anspielen lassen
2–5	Was ist Programmieren: Helfer-Bild, Programmiersprachen, Scratch = Blöcke (optional Roboter-Demo)
5–11	QR-Code einblenden, Klasse beitreten — Account anlegen, Karte ausfüllen
11–14	Remix gemeinsam Schritt für Schritt (Beamer)
14–18	Oberfläche erklären, Mini-Aufgabe Erkunden
18–20	Finger-Bedienung zeigen (Ziehen, Einrasten, lange drücken, Zoom)
20–23	Übung „Hallo!“, Herausforderungen für Schnelle
23–24	Alle speichern, kurz zeigen lassen: Wer hat eine Sprechblase?

Der Block läuft mit der Anmeldung realistisch auf ~24 min. Die Anmeldung ist der einzige Schritt, bei dem die Gruppe wirklich auseinanderläuft: Wer fertig ist, bekommt sofort die Mini-Aufgabe „Erkunden“. Puffer holst du in Block 9 zurück.

Rollen: Fach-Betreuer erklärt und demonstriert am Laptop/Beamer und hält den QR-Code die ganze Zeit auf der Leinwand. Orga-Betreuer verteilt die Karten, hilft beim Anmelden (Nutzername schon vergeben, Passwort zu kurz, Formular nicht abgeschickt), achtet auf WLAN, kontrolliert stichprobenartig, dass Nutzername und Passwort wirklich notiert sind, und sammelt am Ende ein, wer gespeichert hat.

Vorbereiten: Anmeldelink der Klasse erzeugt, als QR-Code auf Folie, Karten gedruckt (Link + Felder für Nutzername, Passwort, Vorname), 3–4 Reserve-Accounts per CSV. Startprojekt „Entenspiel Basis“ enthält: Figures Boot (Kostüme Boot1, Boot2), Ente (Ente1, Ente2), Kacke (Kacke), Bühnenbild background, Klang hit beim Boot, Klänge music_loop und point bei der Bühne. Alle Figuren sichtbar, keine Skripte. Projekt geteilt und im Klassenstudio.

Typische Probleme: QR wird nicht erkannt (Beamer zu hell/dunkel → Link abtippen lassen) · Nutzername schon vergeben · Passwort nicht notiert (fällt erst nach der Pause auf) · Safari zoomt (zwei Finger) · Block nicht eingerastet · falsche Figur ausgewählt (Skript landet bei der Ente) · nicht gespeichert · Remix-Button fehlt (nicht eingeloggt).

Block 2: Das Boot steuern (25 min)

Teilnehmer-Handout. Kästen mit 🎓 sind Hinweise für die Workshop-Leitung und werden vor dem Druck entfernt. Platzhalter für Bilder:  [Bild: ...]

Was du in diesem Block lernst

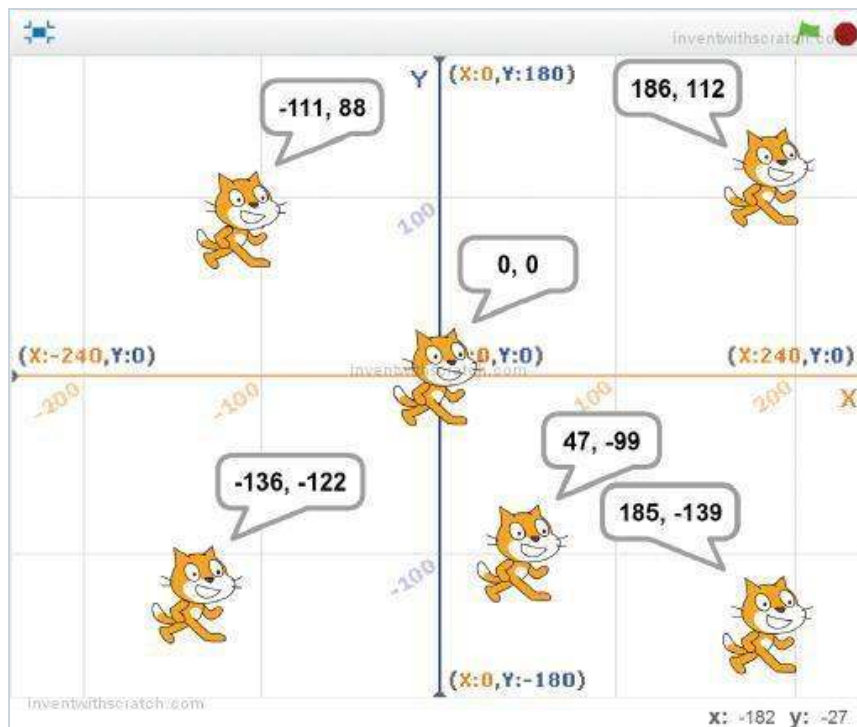
- Wie Scratch Positionen auf der Bühne beschreibt: x und y
- Das Boot an einen festen Startpunkt setzen
- Eine **Endlosschleife** bauen, die ständig prüft, was passiert
- Eine **Bedingung** benutzen: „falls Finger auf der Bühne, dann ...“
- Das Boot deinem Finger folgen lassen

2.1 Wo ist was? Das Koordinatensystem (5 min)

Scratch beschreibt jeden Punkt auf der Bühne mit zwei Zahlen:

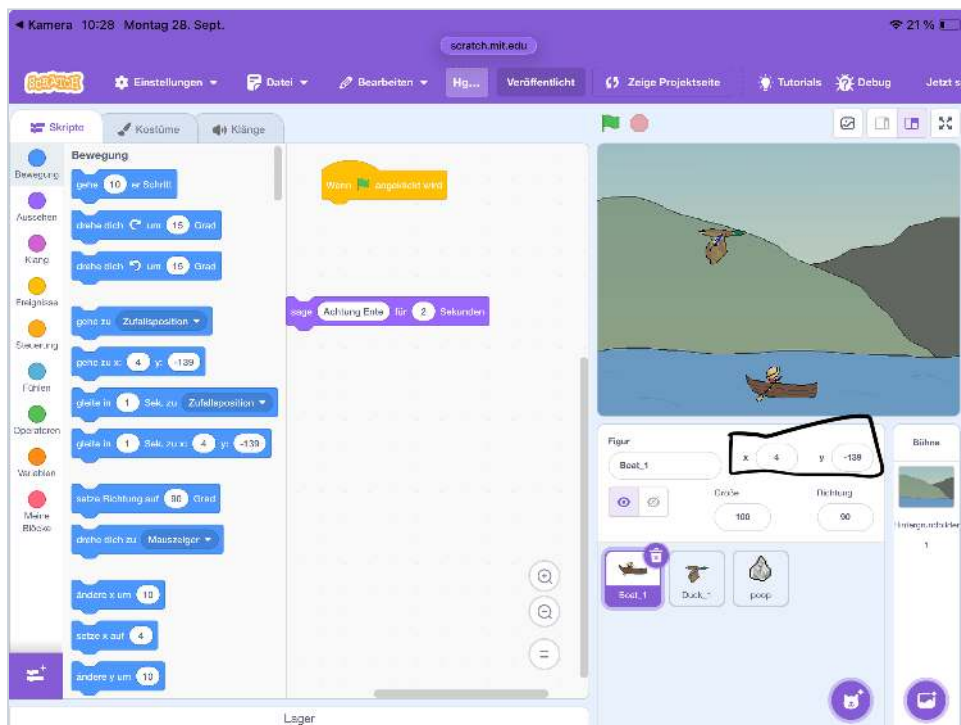
- **x** = links/rechts. Ganz links ist -240, ganz rechts ist 240, die Mitte ist 0.
- **y** = oben/unten. Ganz oben ist 180, ganz unten ist -180, die Mitte ist 0.

Die Bühne ist also 480 breit und 360 hoch. Der Punkt in der Mitte heißt x: 0, y: 0.



Mini-Aufgabe: Koordinaten ablesen (2 min)

1. Wähle das **Boot** in der Figurenliste.
2. Schau unter die Figurenliste: Dort stehen die Felder **x** und **y** des Boots.
3. Zieh das Boot mit dem Finger auf der Bühne herum. Die Zahlen ändern sich live.
4. Zieh das Boot ganz nach unten in die Mitte. Welche Zahlen stehen jetzt da? Ungefähr x: 0 und y: -140?



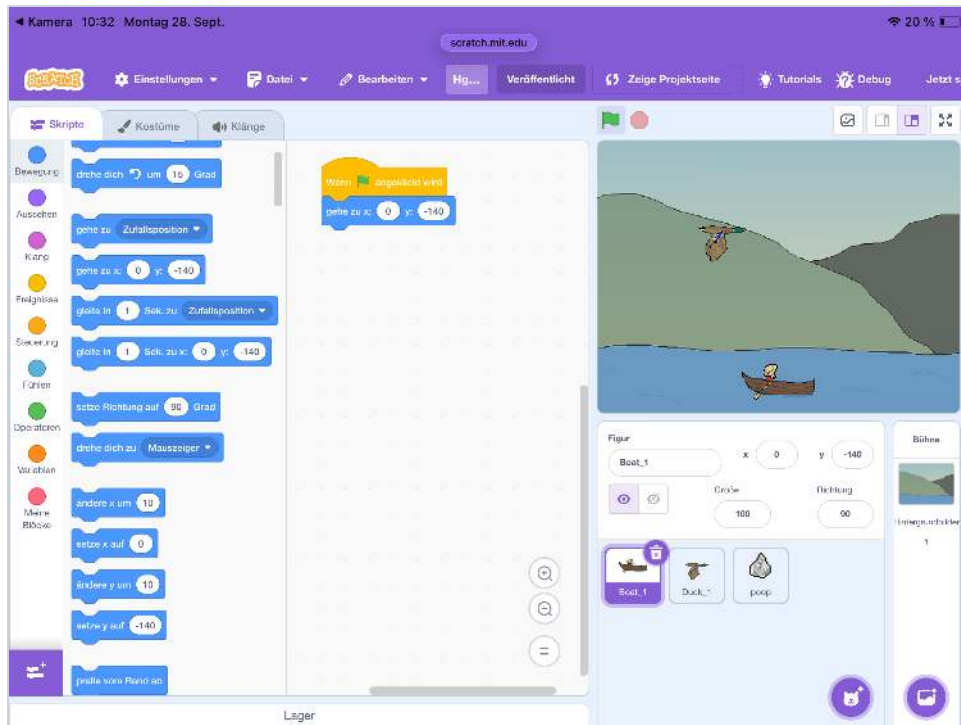
Leitung: Am Beamer kurz das Bühnenbild „Xy-grid“ aus der Bühnenbild-Bibliothek einblenden (Bühne → Bühnenbild wählen → Suche „grid“), Boot herumschieben, Zahlen zeigen, dann Bühnenbild wieder auf „background“ zurück. Teilnehmer machen das nicht mit, nur zuschauen. Am Laptop: Boot mit Maus ziehen = auf dem iPad mit Finger ziehen.

2.2 Das Boot an den Start setzen (3 min)

Nach dem Herumziehen liegt das Boot irgendwo. Beim Start soll es immer unten in der Mitte stehen.

1. Wähle das **Boot**.
2. Kategorie **Bewegung** (blau): Ziehe „**gehe zu x: () y: ()**“ unter deinen gelben Block „Wenn  angeklickt wird“.
 - Falls dort noch der „sage“-Block aus Block 1 hängt: Zieh ihn ab und wirf ihn in die Blockpalette. Oder lass ihn als eigenes kleines Skript liegen.
3. Trage ein: **x: 0** und **y: -140**.
4. Zieh das Boot auf der Bühne irgendwohin. Grüne Flagge. Das Boot springt zurück nach unten.

Wenn  angeklickt wird
gehe zu x: (0) y: (-140)



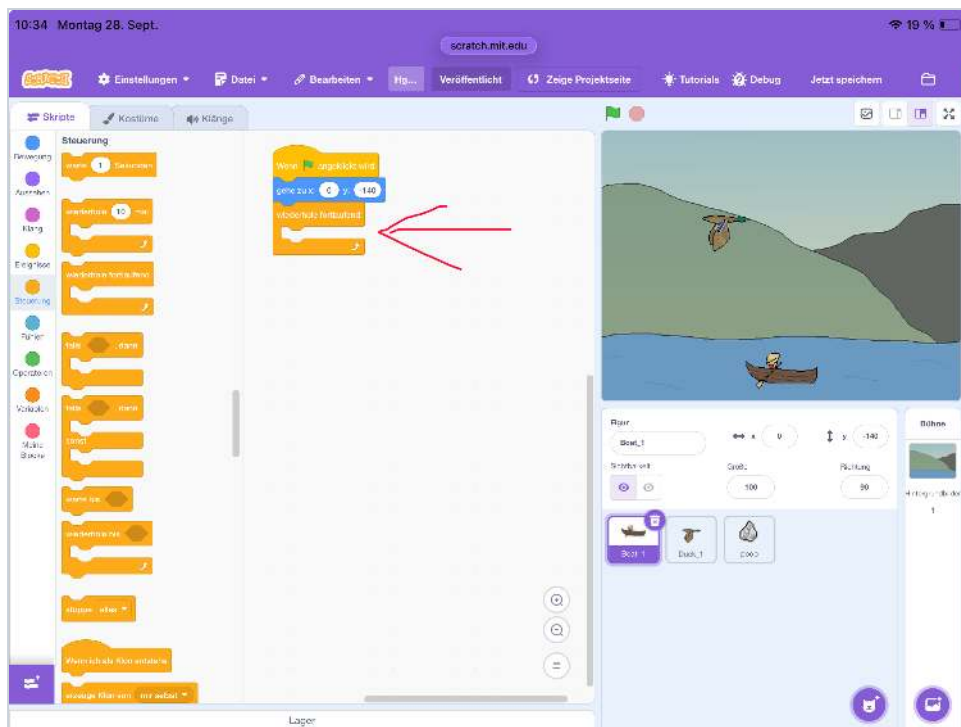
Tip: Wenn du das Boot vorher an die richtige Stelle ziehst, stehen im „gehe zu“-Block in der Palette schon die passenden Zahlen drin.

2.3 Die Endlosschleife (4 min)

Das Spiel soll nicht nur einmal etwas tun, sondern **die ganze Zeit** prüfen, wo dein Finger ist. Dafür gibt es eine Schleife.

1. Kategorie **Steuerung** (orange): Ziehe „**wiederhole fortlaufend**“ unter den „gehe zu“-Block.
2. Der Block ist wie ein Maul. Alles, was du hineinlegst, wird immer und immer wieder ausgeführt, bis das Spiel stoppt.

Wenn  angeklickt wird
gehe zu x: (0) y: (-140)
wiederhole fortlaufend
... (hier kommt gleich etwas hinein)



Probier es aus: Leg zum Test den Block „**nächstes Kostüm**“ (Aussehen, lila) in die Schleife und drücke die Flagge. Das Boot flackert wild zwischen seinen zwei Kostümen. So schnell läuft eine Schleife: etwa 30 Mal pro Sekunde. Nimm den Block danach wieder heraus, den brauchen wir später ordentlicher.

2.4 Übung: Das Boot folgt dem Finger (8 min)

Jetzt kommt die Steuerung. Auf dem iPad gibt es keine Maus, aber Scratch kennt nur „Maus“. Merke dir:

Scratch sagt ...	Auf dem iPad heißt das ...
Maustaste gedrückt?	Dein Finger berührt gerade die Bühne
Maus x-Position	Wo dein Finger links/rechts ist (x-Wert)
Maus y-Position	Wo dein Finger oben/unten ist (y-Wert)

Schritt 1: Die Bedingung

1. Kategorie **Steuerung** (orange): Ziehe „**falls < > dann**“ in die Schleife.
2. Der Block hat oben ein sechseckiges Loch. Da kommt eine Frage hinein, die mit Ja oder Nein beantwortet wird.

Schritt 2: Die Frage

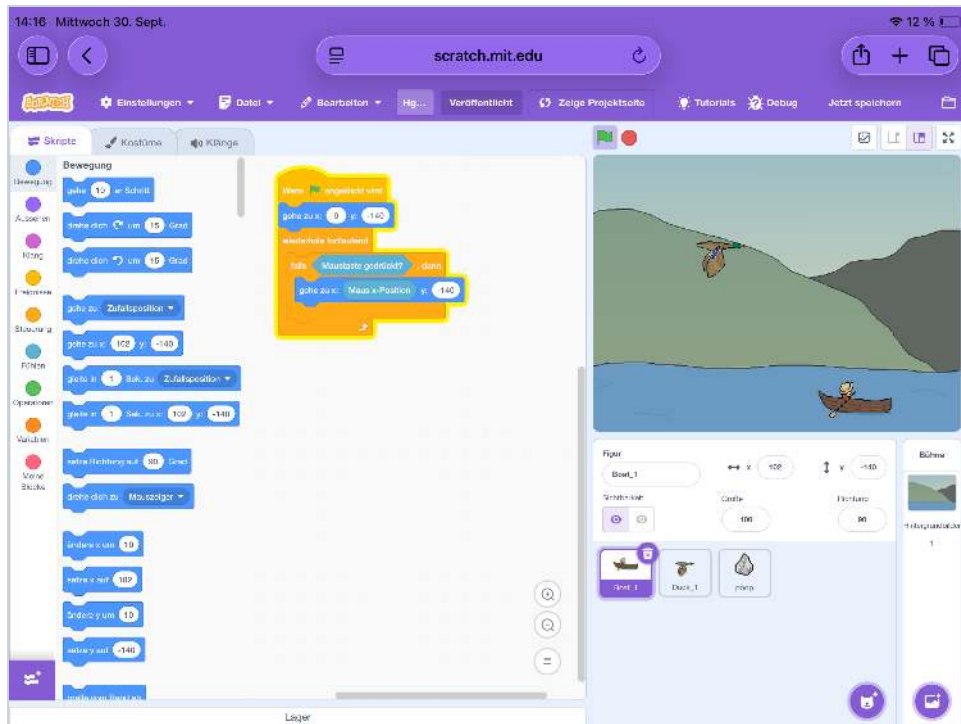
1. Kategorie **Fühlen** (hellblau): Ziehe „**Maustaste gedrückt?**“ in das sechseckige Loch des „falls“-Blocks.

Schritt 3: Die Aktion

1. Kategorie **Bewegung** (blau): Ziehe „**gehe zu x: () y: ()**“ in das Maul des „falls“-Blocks.
2. Trage bei **y** die **-140** ein. Die bleibt fest — das Boot soll ja unten bleiben.
3. Kategorie **Fühlen** (hellblau): Ziehe „**Maus x-Position**“ in das runde Loch bei **x**.

Dein Skript sieht jetzt so aus:

```
Wenn  angeklickt wird
  gehe zu x: (0) y: (-140)
  wiederhole fortlaufend
    falls <Maustaste gedrückt?> dann
      gehe zu x: (Maus x-Position) y: (-140)
```



Schritt 4: Testen

1. Grüne Flagge.
2. Leg deinen Finger auf die Bühne und zieh ihn nach links und rechts. Das Boot fährt mit.
3. Nimm den Finger weg. Das Boot bleibt stehen.

[Bild: Finger auf der Bühne, Boot direkt darunter auf gleicher x-Position, Bewegungspfeil links-rechts]

Schritt 5: Speichern. Jetzt speichern.

Was ist passiert?

Lies das Skript wie einen Satz:

„Wenn die Flagge gedrückt wird: Geh nach unten in die Mitte. Dann prüfe für immer: **Falls** ein Finger auf der Bühne ist, **dann** geh zum x des Fingers — und bleib dabei auf y -140.“

- **Endlosschleife** („wiederhole fortlaufend“): Wiederholt alles in ihrem Maul, bis das Spiel gestoppt wird. Fast jedes Spiel hat so eine Schleife.
- **Bedingung** („falls ... dann“): Prüft eine Frage. Nur wenn die Antwort **Ja** ist, wird das ausgeführt, was drin liegt. Sonst wird es übersprungen.
- **Frage-Blöcke** (sechseckig, wie „Maustaste gedrückt?“): Antworten nur mit Ja oder Nein. Sie passen nur in sechseckige Löcher.
- **Wert-Blöcke** (rund, wie „Maus x-Position“): Liefern eine Zahl oder einen Text. Sie passen nur in runde Löcher.
- Im „gehe zu“-Block steckt bei **x** ein Block, bei **y** eine feste Zahl. Beides ist erlaubt: überall, wo eine Zahl stehen kann, darf auch ein runder Block stecken, der eine Zahl liefert.
- Weil **y** in jedem Durchlauf wieder auf -140 gesetzt wird, bleibt das Boot immer unten — egal wohin du tippst, und auch dann, wenn es vorher jemand nach oben gezogen hat.

[Bild: Gegenüberstellung sechseckiger Block (Frage) und runder Block (Wert), mit passendem Loch daneben]

Herausforderung (wenn du schneller bist)

Such dir eine aus:

1. **Springen statt fahren**: Nimm den „falls“-Block heraus und lass nur den „gehe zu“-Block in der Schleife. Was ist anders, wenn du den Finger wegnimmst und woanders wieder auftippst?

2. **Auch hoch und runter:** Ersetz die -140 bei **y** durch „**Maus y-Position**“ (Fühlen). Jetzt folgt das Boot dem Finger überallhin. Spiel eine Runde so — und überleg, warum wir es trotzdem unten lassen.
3. **Weich fahren (★★★):** Dein Boot springt beim Antippen sofort zum Finger. Es kann auch weich nachziehen — wie ein echtes Boot, das erst Fahrt aufnehmen muss.

Ersetze den „**gehe zu**“-Block durch:

```
ändere x um (((Maus x-Position) - (x-Position)) / (4))
```

Du brauchst dazu zwei grüne **Operatoren**-Blöcke: erst den „**-**“-Block, den setzt du in den „**/**“-Block. Links in den Minus-Block kommt „**Maus x-Position**“ (Fühlen), rechts „**x-Position**“ (Bewegung). Rechts in den Geteilt-Block tippst du die **4**.

Warum das weich wirkt: Der Block rechnet den Abstand zwischen Finger und Boot aus und fährt pro Durchlauf nur ein Viertel davon. Ist der Abstand groß, macht das Boot große Schritte; wird er kleiner, werden auch die Schritte kleiner. Es bremst also von allein ab.

Probier statt der **4** auch mal **2** (schneller, direkter) oder **10** (träge wie ein Lastkahn).

Checkliste Block 2

- ☐ Ich weiß, was x und y auf der Bühne bedeuten.
- ☐ Mein Boot steht beim Start unten in der Mitte (x: 0, y: -140).
- ☐ Ich habe eine „wiederhole fortlaufend“-Schleife im Skript.
- ☐ Das Boot folgt meinem Finger nach links und rechts.
- ☐ Ich habe gespeichert.

Weiter geht's: In Block 3 bringen wir Boot und Ente zum Paddeln und Flattern.

Leitung – Ablauf Block 2 (25 min)

Min	Was
0–5	Koordinaten am Beamer mit Xy-grid zeigen, Mini-Aufgabe ablesen
5–8	„gehe zu x y“ gemeinsam bauen, Reset-Verhalten zeigen
8–12	Endlosschleife erklären, Flacker-Test mit „nächstes Kostüm“
12–20	Steuerung Schritt für Schritt: falls, Maustaste gedrückt, setze x, Maus x-Position. Nach jedem Block kurz warten, bis alle so weit sind
20–23	Testen, Herausforderungen für Schnelle
23–25	„Was ist passiert“ zusammenfassen: Schleife, Bedingung, Frage-Block, Wert-Block. Alle speichern

Vorbereiten: Bühnenbild „Xy-grid“ im eigenen Demo-Projekt schon geladen, damit du nur umschalten musst. Am Laptop-Beamer: „Maus“ ist wirklich Maus. Beim Zeigen immer sagen: „Bei euch ist das der Finger auf der Bühne.“ Ein iPad in der Hand haben und die Fingerbewegung parallel zeigen.

Typische Probleme: Frage-Block „Maustaste gedrückt?“ nicht ins sechseckige Loch eingerastet (liegt lose daneben) · „Maus x-Position“ bei **y** statt bei **x** eingesetzt (Boot klebt dann am unteren Rand und zuckt) · Boot folgt nicht, weil der Finger den Bühnenrand oder eine andere Figur trifft (Boot antippen zählt auch als Bühne, das ist ok) · Schleife vergessen, dann fährt das Boot nur einmal · Skript bei der Ente statt beim Boot gebaut.

Block 3: Animation (10 min)

Teilnehmer-Handout. Kästen mit  sind Hinweise für die Workshop-Leitung und werden vor dem Druck entfernt. Platzhalter für Bilder:  [Bild: ...]

Was du in diesem Block lernst

- Was **Kostüme** sind und wo du sie findest
- Eine Figur zwischen zwei Bildern hin- und herwechseln lassen
- Dass eine Figur **mehrere Skripte gleichzeitig** ausführen kann

3.1 Kostüme anschauen (2 min)

Jede Figur kann mehrere Bilder haben. Scratch nennt sie **Kostüme**. Nur eins ist gerade sichtbar.

1. Wähle das **Boot** in der Figurenliste.
2. Tippe oben links auf den Tab „**Kostüme**“.
3. Du siehst zwei Kostüme: **Boot1** und **Boot2**. Tippe abwechselnd auf beide. Das Boot auf der Bühne wechselt sofort.
4. Schau dir genauso die **Ente** an: **Ente1** und **Ente2**.
5. Zurück auf den Tab „**Skripte**“.

 [Bild: Kostüme-Tab des Boots, beide Kostüme Boot1 und Boot2 links in der Liste, Boot2 ausgewählt]

 [Bild: Kostüme-Tab der Ente, Ente1 und Ente2 nebeneinander, Unterschied der Flügelstellung markiert]

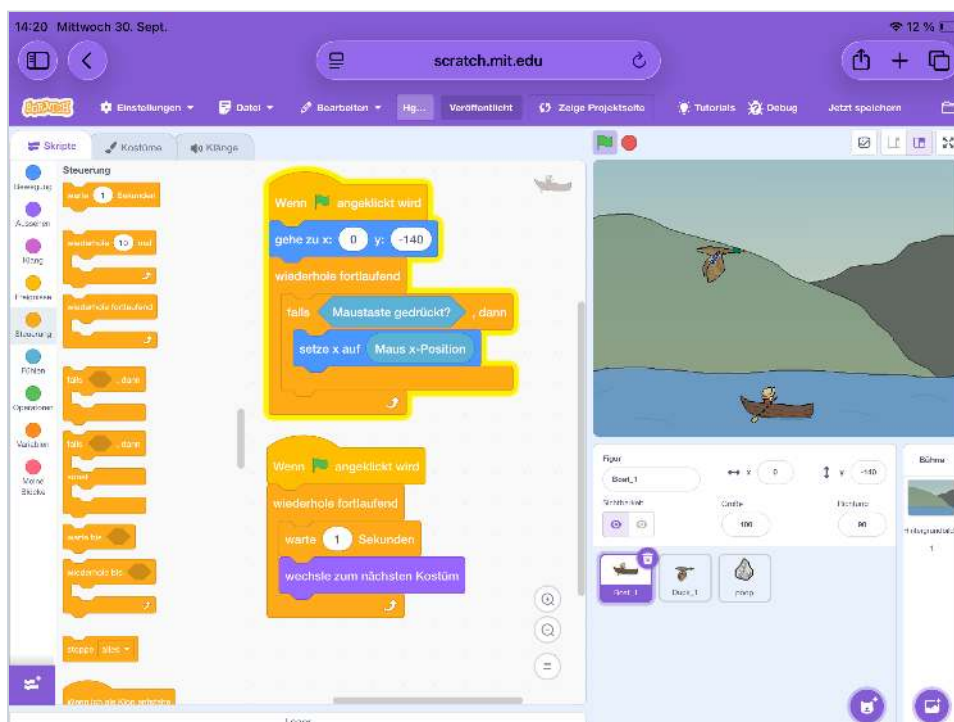
3.2 Übung: Das Boot paddelt (4 min)

Ein Trickfilm ist nichts anderes als schnelles Bilderwechseln. Das machen wir jetzt.

1. Wähle das **Boot**, Tab „**Skripte**“.
2. Ziehe einen **neuen** Block „**Wenn**  **angeklickt wird**“ (Ereignisse) an eine freie Stelle im Skriptbereich. Nicht an das Steuer-Skript hängen, sondern daneben.
3. Darunter „**wiederhole fortlaufend**“ (Steuerung).
4. In die Schleife: „**nächstes Kostüm**“ (Aussehen).
5. Darunter in die Schleife: „**warte (1) Sekunden**“ (Steuerung). Ändere die 1 auf **0.3**.

Wenn  angeklickt wird
wiederhole fortlaufend
nächstes Kostüm
warte (0.3) Sekunden

6. Grüne Flagge. Das Boot paddelt. Und du kannst es gleichzeitig mit dem Finger steuern.
7. Jetzt speichern.



Zahlen ausprobieren: Setz die Wartezeit auf 1, dann auf 0.1. Was passt am besten zu einem Boot?

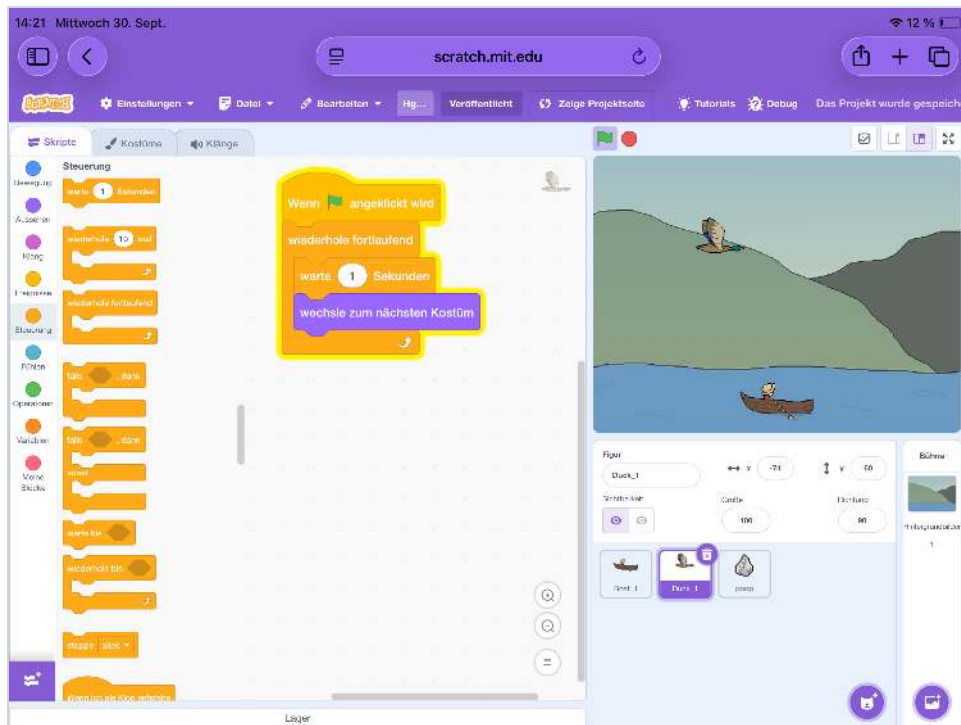
3.3 Übung: Die Ente flattert (2 min)

Das Gleiche für die Ente, nur schneller.

1. Wähle die **Ente**.
2. Baue dasselbe Skript wie beim Boot. Wartezeit: **0.2**.

Wenn  angeklickt wird
wiederhole fortlaufend
nächstes Kostüm
warte (0.2) Sekunden

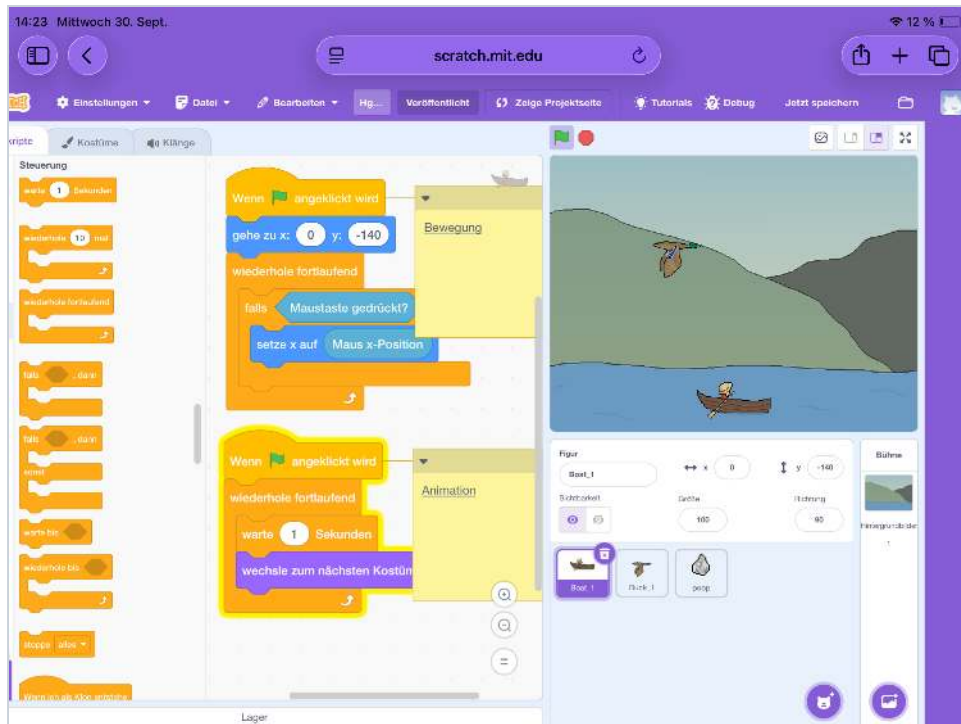
3. Grüne Flagge. Die Ente flattert, das Boot paddelt.



🎓 **Leitung:** Das Kopieren per Ziehen auf die Figurenliste geht auf dem iPad, ist aber fummelig, weil man weit ziehen muss. Zeigen, aber nicht erzwingen. Am Laptop ist es dieselbe Bewegung mit der Maus.

Was ist passiert?

- **Kostüm** = ein Bild einer Figur. „nächstes Kostüm“ springt zum nächsten Bild und nach dem letzten wieder zum ersten.
- **warte** bremst die Schleife. Ohne „warte“ würde die Schleife 30 Mal pro Sekunde wechseln, das Auge sieht nur Flackern.
- Das Boot hat jetzt **zwei Skripte**, die beide mit der grünen Flagge starten. Beide laufen **gleichzeitig**: Eins steuert, eins paddelt. Scratch kann viele Skripte parallel ausführen. So bauen wir das ganze Spiel: Für jede Aufgabe ein eigenes kleines Skript.



Herausforderung (wenn du schneller bist)

Such dir eine aus:

1. **Drittes Kostüm:** Tab „Kostüme“, tippe auf **Boot2** und lange drücken → **„Duplizieren“**. Male mit dem Pinsel etwas dazu, zum Beispiel Wasserspritzer. Jetzt wechselt das Boot durch drei Bilder.
2. **Größe pulsieren:** Bau der Ente ein drittes Skript: Schleife mit **„ändere Größe um (5)“**, **„warte (0.2)“**, **„ändere Größe um (-5)“**, **„warte (0.2)“** (Aussehen).
3. **Nur beim Fahren paddeln:** Beim Boot: Lege „nächstes Kostüm“ und „warte“ in einen **„falls <Maustaste gedrückt> dann“**-Block in der Paddel-Schleife. Jetzt paddelt es nur, wenn dein Finger auf der Bühne liegt.
4. **Farbe:** Schleife mit **„ändere Effekt [Farbe] um (10)“** (Aussehen). Regenbogen-Ente. Mit **„schalte Grafikeffekte aus“** am Skriptanfang wird sie beim Neustart wieder normal.

Checkliste Block 3

- ☐ Ich weiß, wo die Kostüme einer Figur sind.
- ☐ Das Boot wechselt beim Spielen zwischen Boot1 und Boot2.
- ☐ Die Ente flattert.
- ☐ Ich verstehe: Eine Figur kann mehrere Skripte gleichzeitig laufen lassen.
- ☐ Ich habe gespeichert.

Weiter geht's: Pause. Danach lernt die Ente in Block 4 fliegen.

Leitung – Ablauf Block 3 (10 min)

Min	Was
0–2	Kostüme-Tab zeigen, beide Figuren durchtippen
2–6	Paddel-Skript gemeinsam bauen, dabei betonen: NEUES Skript, nicht anhängen. Wartezeiten ausprobieren lassen
6–8	Ente selbstständig, Kopier-Trick zeigen
8–10	Parallelität erklären (zwei Skripte, eine Flagge), speichern, in die Pause

Vorbereiten: Nichts Neues. Kurz prüfen, dass alle das Steuer-Skript aus Block 2 haben, sonst hier nachholen („Entenspiel Stand 02 – Boot steuern“ aus dem Studio remixen lassen).

Typische Probleme: „nächstes Kostüm“ wurde ans Steuer-Skript angehängt statt in ein eigenes Skript (funktioniert zwar, wird aber später unübersichtlich) · „warte“ fehlt oder steht außerhalb der Schleife · Dezimalzahl mit Komma statt Punkt eingegeben: Scratch braucht **0.3**, nicht 0,3 · Kostüm im Kostüme-Tab versehentlich gelöscht (Papierkorb-Symbol): mit „Wiederherstellen“ im Bearbeiten-Menü zurückholen · Tab „Kostüme“ offen gelassen und dann gewundert, wo die Blöcke sind.

Block 4: Die Ente fliegt (25 min)

Teilnehmer-Handout. Kästen mit  sind Hinweise für die Workshop-Leitung und werden vor dem Druck entfernt.

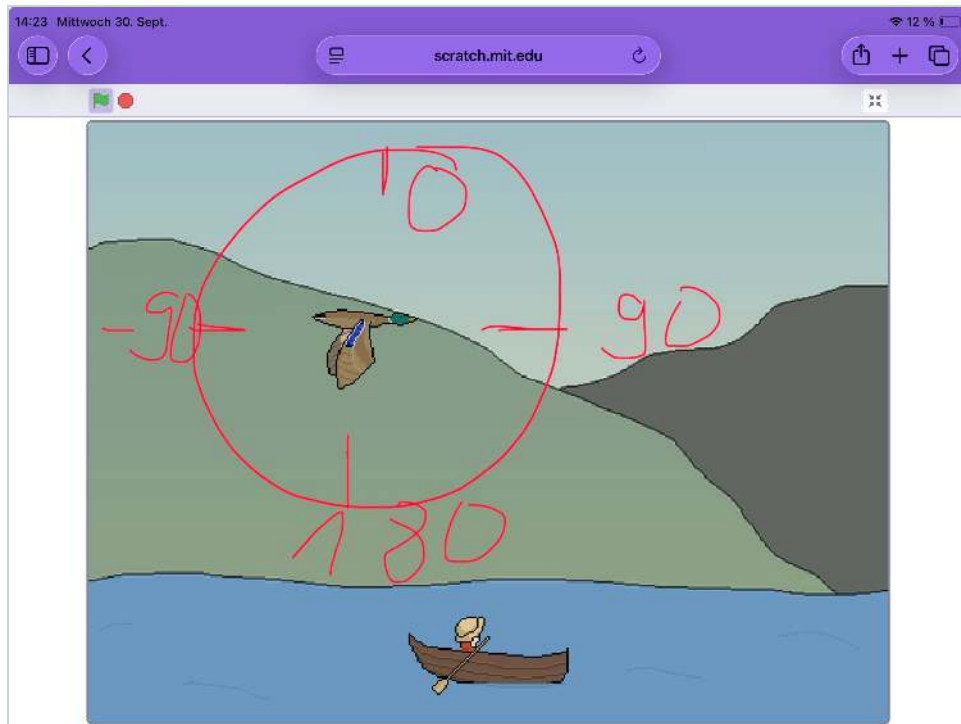
Was du in diesem Block lernst

- Wie eine Figur sich von selbst bewegt: **Richtung** und **Schritte**
- Die Ente am Bühnenrand umdrehen lassen
- Den **Drehtyp** einstellen, damit die Ente nicht kopfüber fliegt
- Was **Zufallszahlen** sind und wozu Spiele sie brauchen

4.1 Richtung und Schritte (5 min)

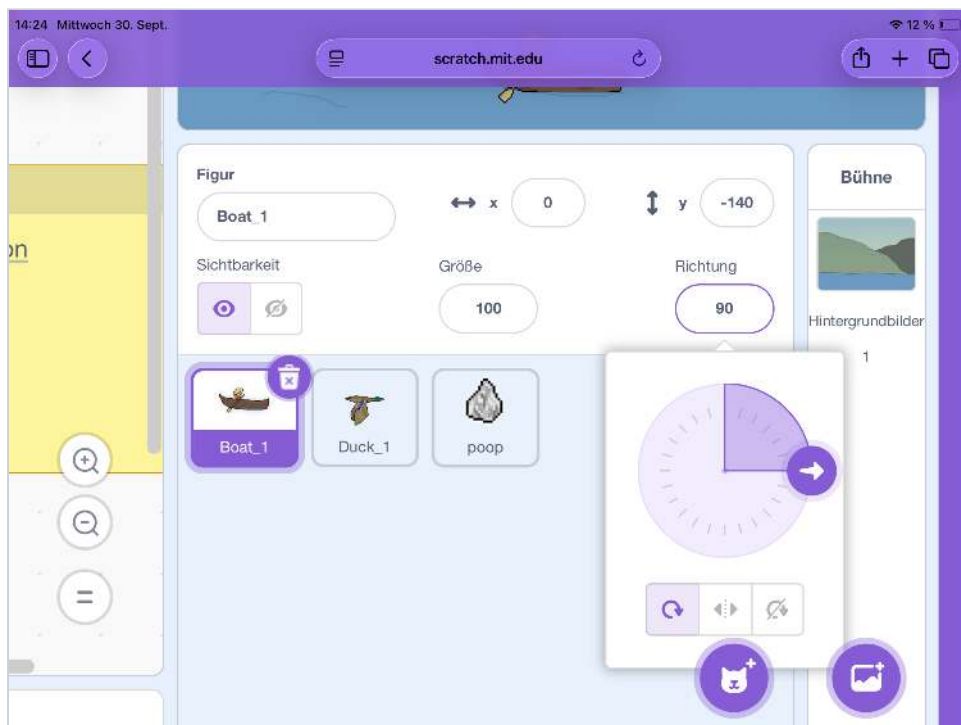
Bis jetzt haben wir Figuren mit **x** und **y** an einen Ort gesetzt. Es gibt noch eine zweite Art, sich zu bewegen: in eine **Richtung** gehen.

- Jede Figur hat eine Richtung, angegeben in Grad wie auf einem Kompass: **90 = rechts**, **-90 = links**, **0 = oben**, **180 = unten**.
- **„gehe (10)er Schritt“** bewegt die Figur 10 Pixel in ihre aktuelle Richtung.



Mini-Aufgabe: Richtung ablesen (2 min)

1. Wähle die **Ente**.
2. Unter der Bühne siehst du das Feld **Richtung**. Tippe darauf. Ein Drehrad erscheint.
3. Dreh die Ente auf 180. Was passiert mit dem Bild auf der Bühne?
4. Stell sie wieder auf **90**.



🎓 Leitung: Das Drehrad-Popup enthält unten auch die drei Drehtyp-Knöpfe (rundum, links-rechts, nicht drehen). Hier noch nicht erklären, kommt in 4.3 als Aha-Moment. Am Laptop: Klick auf Richtung = Tippen auf dem iPad.

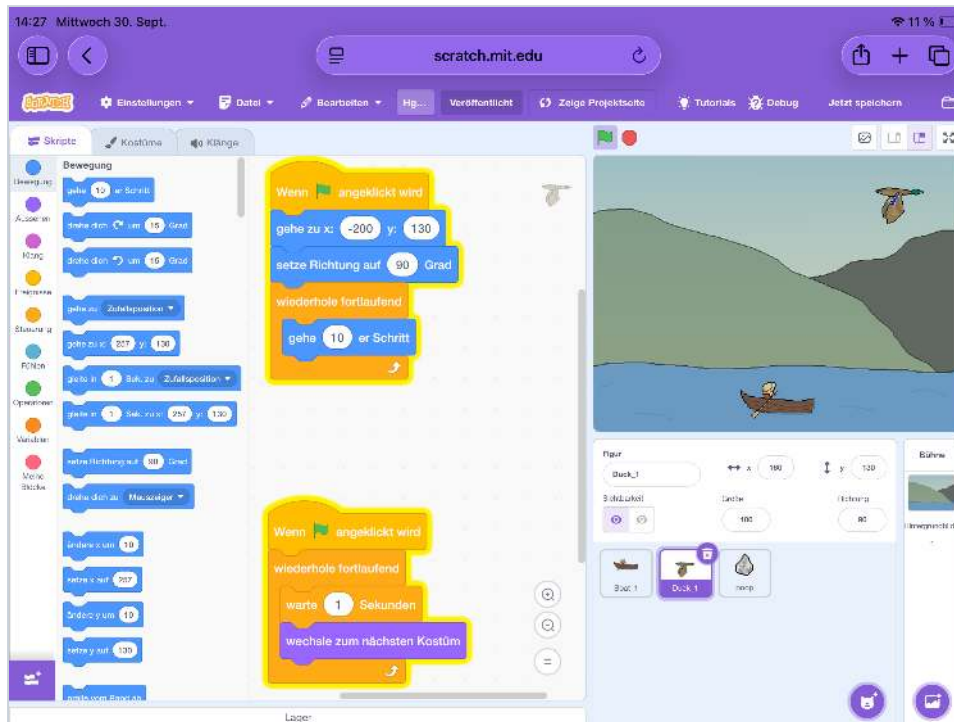
4.2 Übung: Die Ente fliegt los (6 min)

1. Wähle die **Ente**. Du hast schon das Flutter-Skript. Jetzt kommt ein **zweites** Skript daneben.
2. „**Wenn angeklickt wird**“ (Ereignisse) an eine freie Stelle.
3. „**gehe zu x: (-200) y: (130)**“ (Bewegung) darunter. Die Ente startet oben links.

4. „**setze Richtung auf (90)**“ (Bewegung) darunter. Sie schaut nach rechts.
5. „**wiederhole fortlaufend**“ (Steuerung) darunter.
6. In die Schleife: „**gehe (10)er Schritt**“ (Bewegung). Ändere die 10 auf **4**.

Wenn  angeklickt wird
 gehe zu x: (-200) y: (130)
 setze Richtung auf (90)
 wiederhole fortlaufend
 gehe (4)er Schritt

7. Grüne Flagge. Die Ente fliegt nach rechts, bleibt am Rand kleben und kommt nicht zurück.

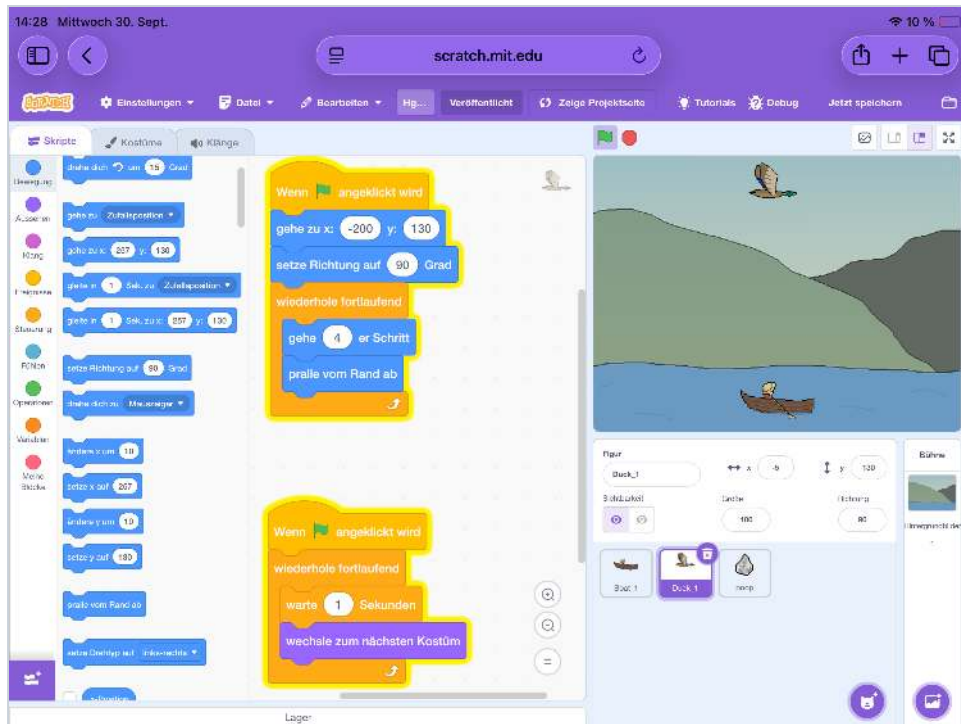


Warum bleibt sie hängen? Scratch lässt Figuren nicht ganz aus der Bühne verschwinden. Aber umdrehen muss sie selbst. Das machen wir jetzt.

4.3 Übung: Am Rand umdrehen (6 min)

Schritt 1: Abprallen

1. „**pralle vom Rand ab**“ (Bewegung) in die Schleife, unter „gehe (4)er Schritt“.
2. Grüne Flagge. Die Ente fliegt hin und her. Aber schau genau hin: Auf dem Rückweg fliegt sie **kopfüber**!

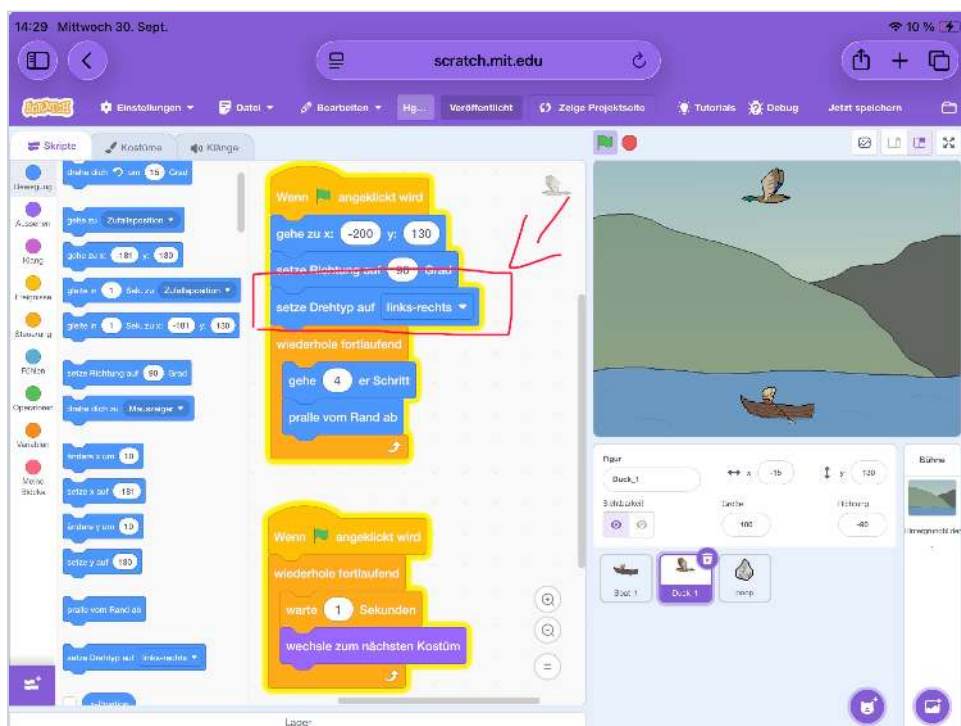


Schritt 2: Drehtyp

Beim Abprallen dreht Scratch die Ente um 180 Grad, wie ein Ball, der sich dreht. Wir wollen, dass sie nur **gespiegelt** wird.

1. „**setze Drehtyp auf [links-rechts]**“ (Bewegung) unter „setze Richtung auf (90)“, also **vor** der Schleife.
2. Grüne Flagge. Jetzt dreht die Ente sich richtig um.

Wenn angeklickt wird
 gehe zu x: (-200) y: (130)
 setze Drehtyp auf [links-rechts]
 setze Richtung auf (90)
 wiederhole fortlaufend
 gehe (4)er Schritt
 pralle vom Rand ab



3. Jetzt speichern.

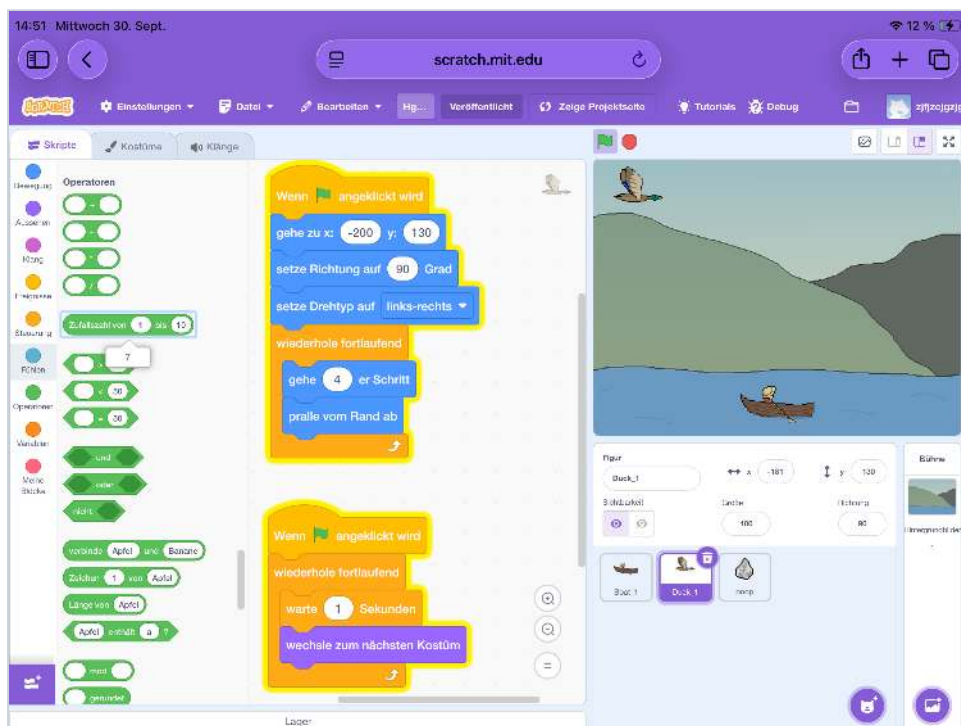
Was ist passiert?

- **Richtung** sagt, wohin die Figur schaut. „**gehe ()er Schritt**“ bewegt sie in diese Richtung. Beides zusammen ist wie ein Auto: Lenken und Gas geben.
- „**pralle vom Rand ab**“ prüft bei jedem Durchlauf: Berühre ich den Rand? Wenn ja, dreh die Richtung um. Ein Block, der eine Bedingung schon eingebaut hat.
- **Drehtyp** bestimmt, wie das Bild bei einer Richtungsänderung gezeichnet wird:
 - **rundum**: Bild dreht sich mit, wie eine Rakete. Bei 180 Grad steht es auf dem Kopf.
 - **links-rechts**: Bild wird nur gespiegelt. Richtig für Tiere, Autos, Boote.
 - **nicht drehen**: Bild bleibt immer gleich.
- Die Ente hat jetzt **zwei Skripte**: Flattern (Block 3) und Fliegen (jetzt). Beide laufen gleichzeitig.

4.4 Zufall (5 min)

Ein Spiel, das immer gleich abläuft, ist langweilig. Spiele brauchen **Zufall**.

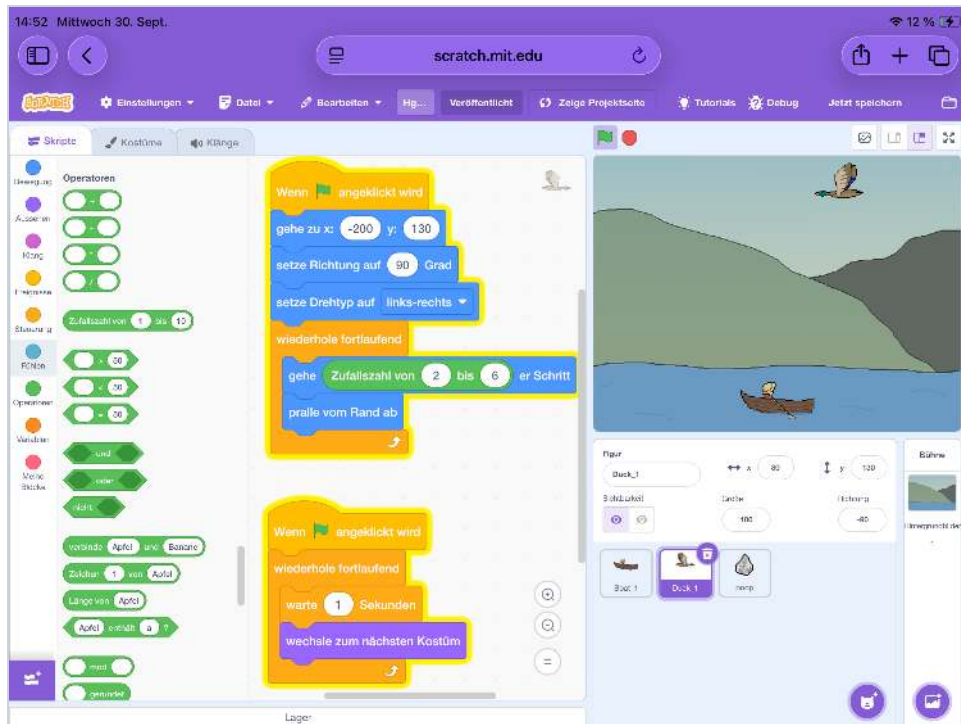
1. Kategorie **Operatoren** (grün): Finde den Block „**Zufallszahl von (1) bis (10)**“. Er ist rund, also ein Wert-Block.
2. Tippe ihn ein paar Mal direkt in der Palette an. Jedes Mal erscheint eine andere Zahl in einer Blase.



Mini-Aufgabe: Zufälliges Tempo (3 min)

1. Ziehe „**Zufallszahl von (1) bis (10)**“ in das runde Loch von „**gehe (4)er Schritt**“. Die 4 wird ersetzt.
2. Ändere auf **von 2 bis 6**.
3. Grüne Flagge. Die Ente ruckelt, weil sie in jedem Durchlauf unterschiedlich weit springt.

gehe (Zufallszahl von (2) bis (6))er Schritt



4. Das sieht nicht schön aus. Bau die **4** wieder ein (Zufallszahl-Block herausziehen und in die Palette werfen). Zufall brauchen wir gleich an einer besseren Stelle.

Merke: Wert-Blöcke kannst du in jedes runde Loch stecken, auch anstelle einer festen Zahl. So werden aus starren Skripten lebendige.

Leitung: Der Ruckel-Effekt ist Absicht: zeigt, dass Zufall pro Schleifendurchlauf neu gewürfelt wird (30 Mal pro Sekunde). Wer es behalten will, darf. Die saubere Variante (Tempo nach jedem Abprall neu würfeln) ist Herausforderung 2. In Block 7 wird das Tempo dann per Variable gesteuert.

Herausforderung (wenn du schneller bist)

Such dir eine aus:

- Wackelflug:** In die Flug-Schleife zusätzlich „**ändere y um (Zufallszahl von (-2) bis (2))**“ (Bewegung + Operatoren). Die Ente wackelt beim Fliegen auf und ab.
- Unregelmäßig schnell:** Setze „**warte (Zufallszahl von (0) bis (0.05)) Sekunden**“ (Steuerung + Operatoren) in die Flug-Schleife. Die Ente wird mal schneller, mal langsamer. Ein festes neues Tempo nach jedem Abprall braucht eine Variable, das kommt in Block 7.
- Zufälliger Start:** Ersetze im „gehe zu x: y:“ das **y** durch „**Zufallszahl von (100) bis (150)**“. Bei jedem Start fliegt die Ente auf anderer Höhe.
- Quaken:** Tab „Klänge“ → Lautsprecher-Symbol unten links → Bibliothek → Suche „Duck“. Dann in der Flug-Schleife: „**falls <wird [Rand] berührt?> dann spiele Klang [Duck]**“. Sie quakt beim Umdrehen.

Checkliste Block 4

- ☐ Die Ente startet oben links und fliegt nach rechts.
- ☐ Sie dreht am Rand um.
- ☐ Sie fliegt auf dem Rückweg **nicht** kopfüber (Drehtyp links-rechts).
- ☐ Ich weiß, was der Zufallszahl-Block macht und wo er hinpasst.
- ☐ Ich habe gespeichert.

Weiter geht's: In Block 5 fängt die Ente an zu kacken. Dafür lernen wir Klone kennen.

🎓 Leitung – Ablauf Block 4 (25 min)

Min	Was
0–5	Richtung als Kompass erklären, Drehrad zeigen, Mini-Aufgabe
5–11	Flug-Skript gemeinsam bis „gehe 4 Schritte“. Hängenbleiben am Rand als Frage stellen: „Was fehlt?“
11–17	„pralle vom Rand ab“ hinzufügen. Kopfüber-Effekt entdecken lassen, dann Drehtyp erklären und einbauen. Speichern
17–22	Zufallszahl: in Palette antippen, ins Tempo stecken, Ruckeln sehen, wieder raus. Herausforderungen
22–25	„Was ist passiert“ zusammenfassen: Richtung, Abprallen, Drehtyp, Wert-Block in rundes Loch. Speichern

Vorbereiten: Demo-Projekt mit Stand nach Block 3. Am Laptop-Beamer den Kopfüber-Moment groß zeigen (Bühne auf groß stellen), das ist der Lacher des Blocks und bleibt hängen. Herausforderung 4 (Klang „Duck“) vorher in der Bibliothek testen, Name der Klangdatei prüfen.

Typische Probleme: „setze Drehtyp“ in die Schleife gelegt statt davor (funktioniert, ist aber unnötig) · Flug-Skript ans Flatter-Skript angehängt: dann flattert die Ente nicht mehr, weil „warte 0.2“ auch das Fliegen bremst. Lösung: zwei getrennte Skripte · Richtung von Hand auf 180 gelassen, Ente fliegt nach unten · „gehe zu x: -200“ mit Bindestrich statt Minus getippt (auf dem iPad-Nummernpad die Minus-Taste zeigen) · Zufallszahl in „gehe zu x: y:“ statt in „gehe ()er Schritt“ gesteckt · Ente zu groß, berührt sofort den Rand und zittert: Größe unter der Figurenliste auf 60–80 % setzen.

Block 5: Es kackt! (30 min)

Teilnehmer-Handout. Kästen mit 🎓 sind Hinweise für die Workshop-Leitung und werden vor dem Druck entfernt.

Was du in diesem Block lernst

- Was ein **Klon** ist und warum wir ihn brauchen
- Die Ente lässt regelmäßig eine Kacke fallen
- Die Kacke fällt nach unten und verschwindet am Bühnenrand
- Zahlen vergleichen mit dem grünen Operator-Block „<“

5.1 Das Problem: Viele Kacken, aber nur eine Figur (3 min)

Schau in die Figurenliste: Es gibt genau **eine** Figur „Kacke“. Im Spiel sollen aber mehrere Kacken gleichzeitig fallen, um eine echte Herausforderung fuer den Spieler darzustellen.

Wir könnten die Figur zehnmal duplizieren. Das wäre viel Arbeit, und nach 10 Kacken wäre Schluss.

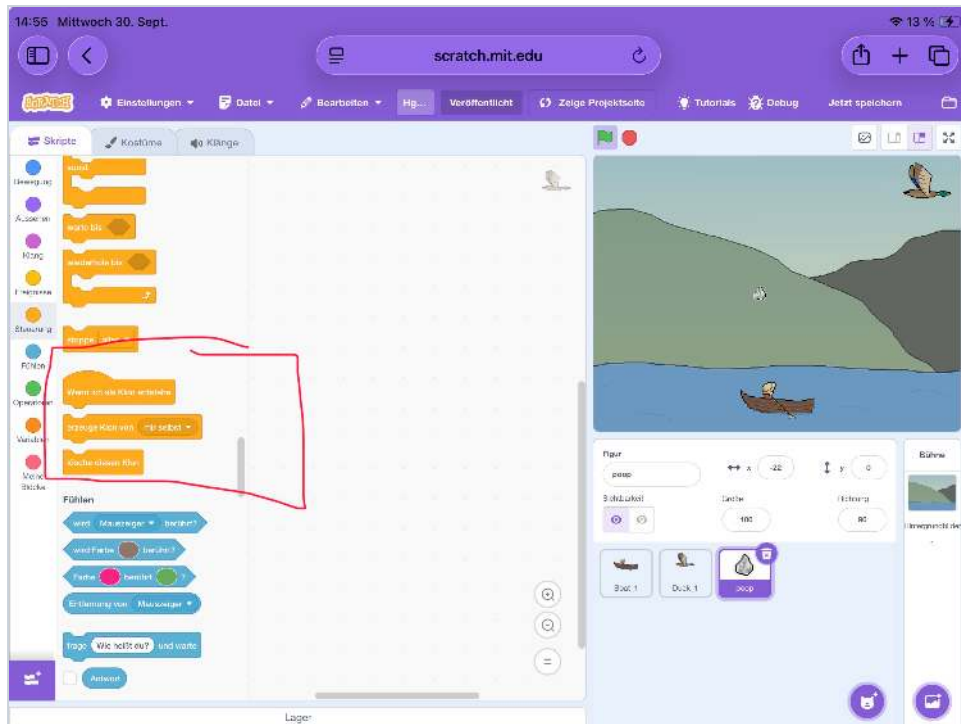
Scratch hat dafür einen Trick: **Klone**.

Ein Klon ist eine Kopie einer Figur, die das Spiel selbst erzeugt, während es läuft.

Stell dir die Figur „Kacke“ wie einen **Stempel** vor. Der Stempel selbst bleibt versteckt in der Schublade. Jedes Mal, wenn die Ente kacken soll, drückt sie den Stempel einmal auf die Bühne. Es entsteht ein Abdruck: ein Klon. Der Abdruck lebt sein eigenes Leben, fällt nach unten und löst sich am Ende auf.

Drei Blöcke gehören zu Klonen (alle in **Steuerung**, orange, ganz unten in der Liste):

Block	Was er tut
„erzeuge Klon von (...)“	Stempelt einen neuen Klon. Kann jede Figur aufrufen, auch für eine andere Figur.
„Wenn ich als Klon entstehe“	Hut-Block. Startet nur für den neuen Klon, nicht für das Original.
„lösche diesen Klon“	Der Klon verschwindet für immer.



Leitung: Stempel-Bild am Beamer wörtlich nehmen: ein echter Stempel oder ein Post-it-Block in der Hand hilft. Betonen: Das Original bleibt immer versteckt, die Klone sind die sichtbaren Kacken. Das ist der häufigste Denkfehler in diesem Block.

5.2 Übung: Die Ente stempelt (5 min)

Wähle in der Figurenliste die **Ente**. Sie hat schon zwei Skripte (Fliegen und Flattern). Wir bauen ein **drittes** daneben.

Schritt 1: Hut-Block

1. Kategorie **Ereignisse** (gelb) → „**Wenn** **angeklickt wird**“ an eine freie Stelle im Skriptbereich ziehen.

Schritt 2: Endlos wiederholen

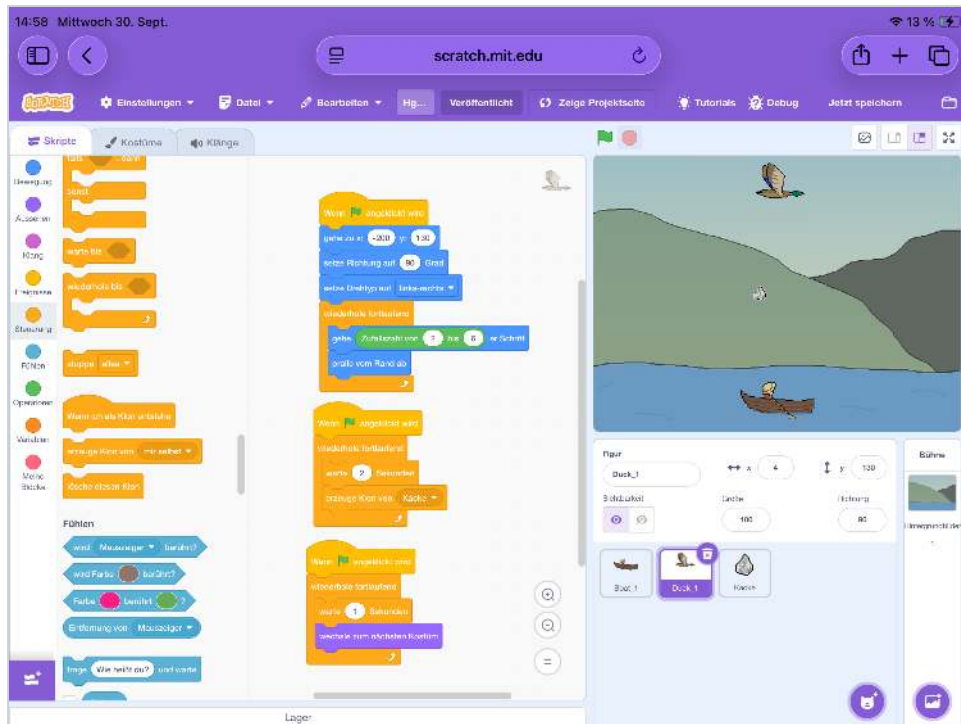
1. Kategorie **Steuerung** (orange) → „**wiederhole fortlaufend**“ darunter einrasten.

Schritt 3: Warten und stempeln

1. **Steuerung** → „**warte (1) Sekunden**“ in die Schleife ziehen. Ändere die 1 auf **2**.
2. **Steuerung** → ganz nach unten scrollen → „**erzeuge Klon von (mir selbst)**“ unter den warte-Block in die Schleife.
3. Tippe auf das Dropdown „**mir selbst**“ und wähle „**Kacke**“.

Dein drittes Enten-Skript:

```
Wenn  angeklickt wird
wiederhole fortlaufend
  warte (2) Sekunden
  erzeuge Klon von [Kacke]
```



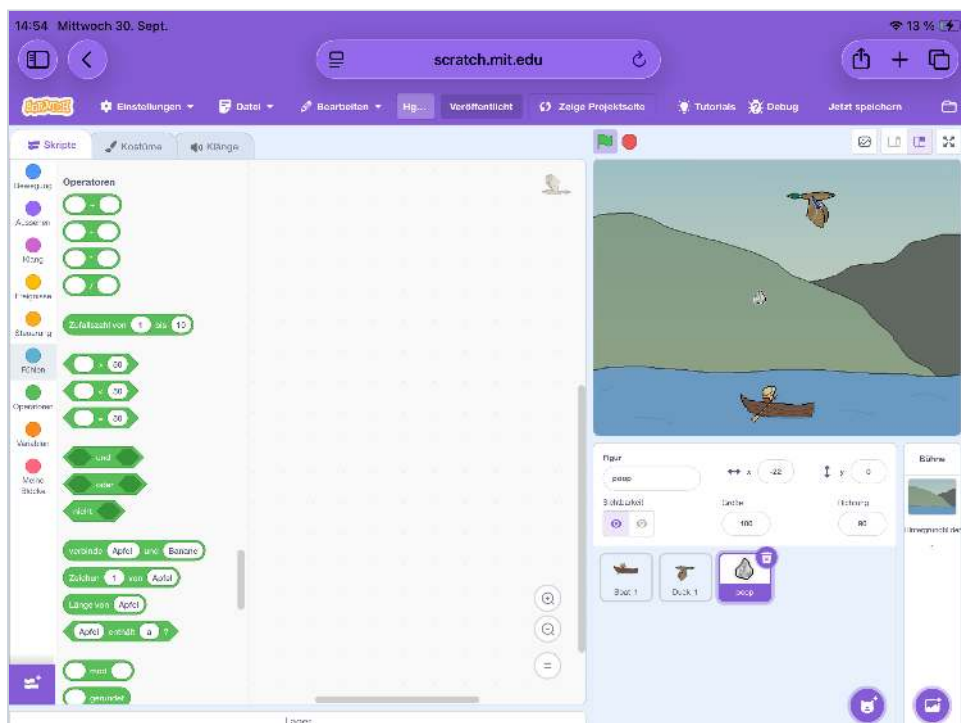
Schritt 4: Testen

Grüne Flagge. Was passiert? Wahrscheinlich siehst du **nichts Neues**. Die Klone entstehen zwar alle 2 Sekunden, aber sie liegen unsichtbar genau auf dem Original. Sie wissen noch nicht, was sie tun sollen. Das ändern wir jetzt.

Leitung: Dropdown „mir selbst“ → „Kacke“ ist auf dem iPad ein Antippen, danach Liste. Wer die Klon-Blöcke nicht findet: In der Steuerung-Palette ganz nach unten scrollen. Bei der Demo am Laptop: Palette mit Mausrad scrollen = auf dem iPad mit dem Finger in der Palette nach oben wischen.

5.3 Übung: Das Original versteckt sich (3 min)

Wähle die **Kacke** in der Figurenliste. Sie liegt noch mitten auf der Bühne und hat kein einziges Skript — so sieht sie jetzt aus:



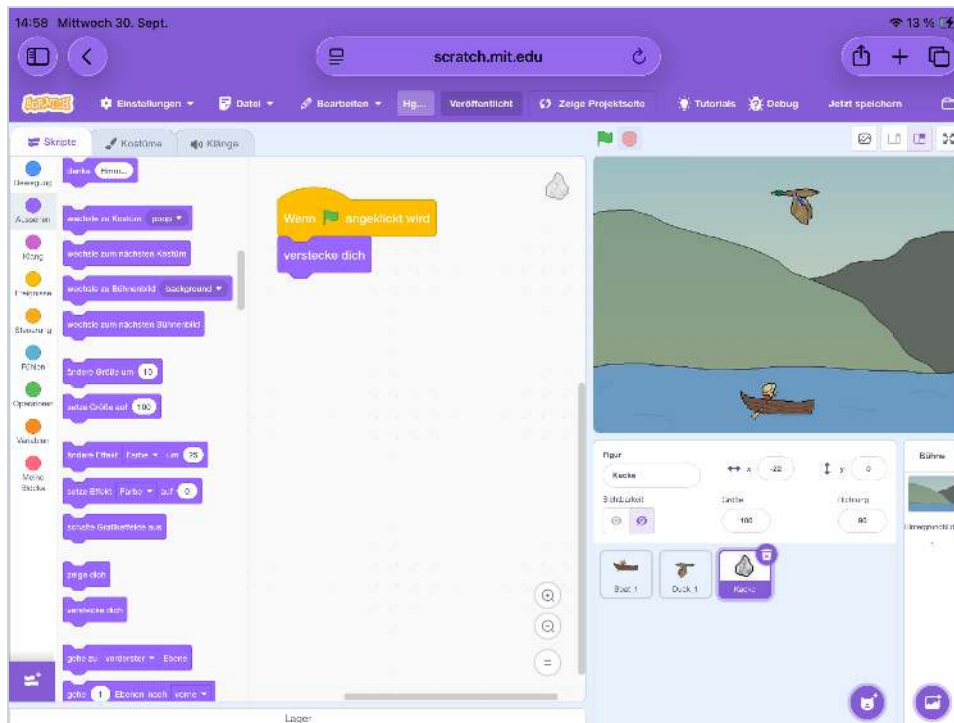
Das Original soll niemand sehen. Es ist nur der Stempel.

1. **Ereignisse** → „**Wenn**  **angeklickt wird**“.

2. **Aussehen** (lila) → „**verstecke dich**“ darunter.

Wenn  angeklickt wird
verstecke dich

Grüne Flagge: Die Kacke verschwindet von der Bühne. Gut so.



5.4 Übung: Der Klon fällt (10 min)

Bleib bei der **Kacke**. Jetzt bekommt der Klon seine Anweisungen.

Schritt 1: Der Klon-Hut

1. **Steuerung** → „**Wenn ich als Klon entstehe**“ an eine freie Stelle ziehen.

Schritt 2: Zur Ente springen und auftauchen

1. **Bewegung** (blau) → „**gehe zu (Zufallsposition)**“ darunter. Dropdown antippen → „**Ente**“ wählen.
2. **Bewegung** → „**ändere y um (10)**“ darunter. Ändere 10 auf **-20** (minus zwanzig). So startet die Kacke ein Stück unter der Ente, nicht in ihrem Bauch.
3. **Aussehen** → „**zeige dich**“ darunter.

Schritt 3: Fallen, bis unten

1. **Steuerung** → „**wiederhole bis <>**“ darunter. Der Block hat ein leeres sechseckiges Loch. Da kommt eine Bedingung rein.
2. **Operatoren** (grün) → „**() < (50)**“ in das sechseckige Loch ziehen.
3. **Bewegung** → „**y-Position**“ (der runde Block, weiter unten in der Palette) in das **linke** Loch des „<“-Blocks ziehen.
4. Tippe ins rechte Feld, ändere 50 auf **-170**.
5. **Bewegung** → „**ändere y um (10)**“ in die Schleife. Ändere 10 auf **-5**.

Schritt 4: Aufräumen

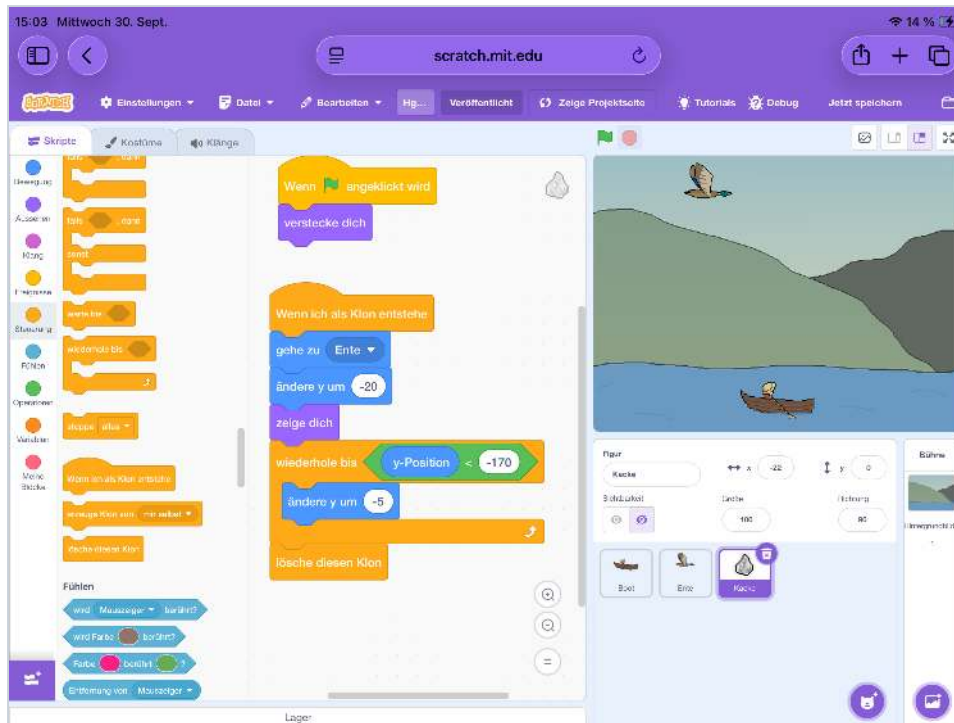
1. **Steuerung** → „**lösche diesen Klon**“ unter die Schleife (nicht hinein!).

Dein Klon-Skript:

```

Wenn ich als Klon entstehe
gehe zu [Ente]
ändere y um (-20)
zeige dich
wiederhole bis <(y-Position) < (-170)>
  ändere y um (-5)
lösche diesen Klon

```



Schritt 5: Testen

Grüne Flagge. Alle 2 Sekunden fällt eine Kacke von der Ente. Unten verschwindet sie. Steuere das Boot ein bisschen. Noch passiert nichts beim Treffer. Das kommt in Block 6.

Speichern!

Leitung: Das ist die längste Übung des Tages. Am Beamer Schritt für Schritt bauen und nach Schritt 2 einmal testen lassen (Klone erscheinen an der Ente, fallen noch nicht). Dann weiter. Häufige Fehler: „lösche diesen Klon“ landet in der Schleife → Kacke verschwindet sofort nach einem Schritt. „ändere y um -5“ landet unter statt in der Schleife → Kacke bleibt hängen. Minuszeichen vergessen → Kacke steigt nach oben. Auf dem iPad öffnet das Zahlenfeld ein Nummernpad, das Minus ist links unten.

5.5 Was ist passiert? (4 min)

Klone

- Die **Ente** stempelt: „erzeuge Klon von Kacke“. Das darf sie, obwohl sie nicht die Kacke ist.
- Jeder neue Klon startet **sein eigenes** Skript „Wenn ich als Klon entstehe“. Fünf Klone = fünf Skripte gleichzeitig.
- Der Klon erbt alles vom Original: Kostüm, Größe, Position, **auch das Versteckt-Sein**. Deshalb brauchen wir „zeige dich“.
- „lösche diesen Klon“ gilt nur für den einen Klon. Die anderen fallen weiter.
- Das Original bleibt die ganze Zeit versteckt. Es ist nur die Vorlage.

Vergleichen mit „<“

Der grüne Block „<“ ist eine Frage: „Ist die linke Zahl kleiner als die rechte?“ Die Antwort ist **wahr** oder **falsch**.

- „y-Position“ ist die aktuelle Höhe des Klons. Oben ist +180, unten -180.
- $(y\text{-Position}) < (-170)$ heißt: „Bin ich schon fast ganz unten?“

- „wiederhole bis" wiederholt, **bis die Antwort wahr ist**. Also: Fallen, fallen, fallen ... unten angekommen → aufhören → löschen.

Herausforderung (wenn du schneller bist)

Such dir eine aus:

1. **Schneller Fall:** Ändere ändere y um (-5) auf -8 oder -12. Wie fühlt sich das Spiel an? Wie hoch darf die Zahl werden, bis es unfair wird?
2. **Häufiger kacken:** Ändere bei der Ente warte (2) Sekunden auf 1 oder 0.5. Ist das noch schaffbar?
3. **Wackel-Kacke:** In der Fall-Schleife zusätzlich „ändere x um ()" mit **Operatoren** → „Zufallszahl von (-2) bis (2)" im Loch. Die Kacke trudelt beim Fallen.
4. **Drehende Kacke:** In der Fall-Schleife **Bewegung** → „drehe dich um (5) Grad". Achtung: Dann muss der Klon am Anfang „setze Richtung auf (90)" bekommen, sonst startet jeder Klon schief.

Checkliste Block 5

- ☐ Ich kann in einem Satz erklären, was ein Klon ist.
- ☐ Die Ente erzeugt alle 2 Sekunden einen Klon der Kacke.
- ☐ Das Original der Kacke ist versteckt.
- ☐ Die Kacke-Klone fallen von der Ente nach unten und verschwinden.
- ☐ Ich habe auf „Jetzt speichern" getippt.

Weiter geht's: In Block 6 tut ein Treffer weh: Leben zählen, Boot blinkt, Game Over.

Leitung – Ablauf Block 5 (30 min)

Min	Was
0–3	Problem zeigen: eine Figur, viele Kacken. Stempel-Bild einführen, drei Klon-Blöcke am Beamer zeigen
3–8	Übung 5.2 Ente stempelt, gemeinsam bauen, testen (nichts sichtbar, absichtlich)
8–11	Übung 5.3 Original verstecken
11–21	Übung 5.4 Klon fällt, Schritt für Schritt am Beamer, Zwischentest nach Schritt 2
21–25	„Was ist passiert?": Klone, „<"-Block, „wird (Rand) berührt?"-Falle
25–29	Herausforderungen, Orga-Betreuer geht herum und prüft die Checkliste
29–30	Alle speichern

Vorbereiten: Zwischenstand „Entenspiel Stand 04 – Ente fliegt" im Klassenstudio für alle, die in Block 4 nicht fertig wurden (Remix, dann hier einsteigen). Eigenes Projekt mit fertigem Block 5 zum Vorzeigen. Stempel oder Post-it-Block als Requisite.

Beamer über Laptop: Dropdown öffnen = Klick auf den Pfeil, auf dem iPad Antippen. Palette scrollen = Mausrad, auf dem iPad in der Palette wischen. Minus im Zahlenfeld: am iPad Nummernpad links unten.

Typische Probleme: „lösche diesen Klon" in der Schleife statt darunter · „ändere y um -5" unter statt in der Schleife · Minus vergessen (Kacke steigt) · Klon-Skript bei der Ente statt bei der Kacke gebaut · „gehe zu" auf „Zufallsposition" gelassen · „zeige dich" vergessen (Klone unsichtbar) · „erzeuge Klon von mir selbst" nicht auf Kacke umgestellt (Enten vermehren sich).

Block 6: Treffer & Leben (25 min)

Teilnehmer-Handout. Kästen mit  sind Hinweise für die Workshop-Leitung und werden vor dem Druck entfernt.

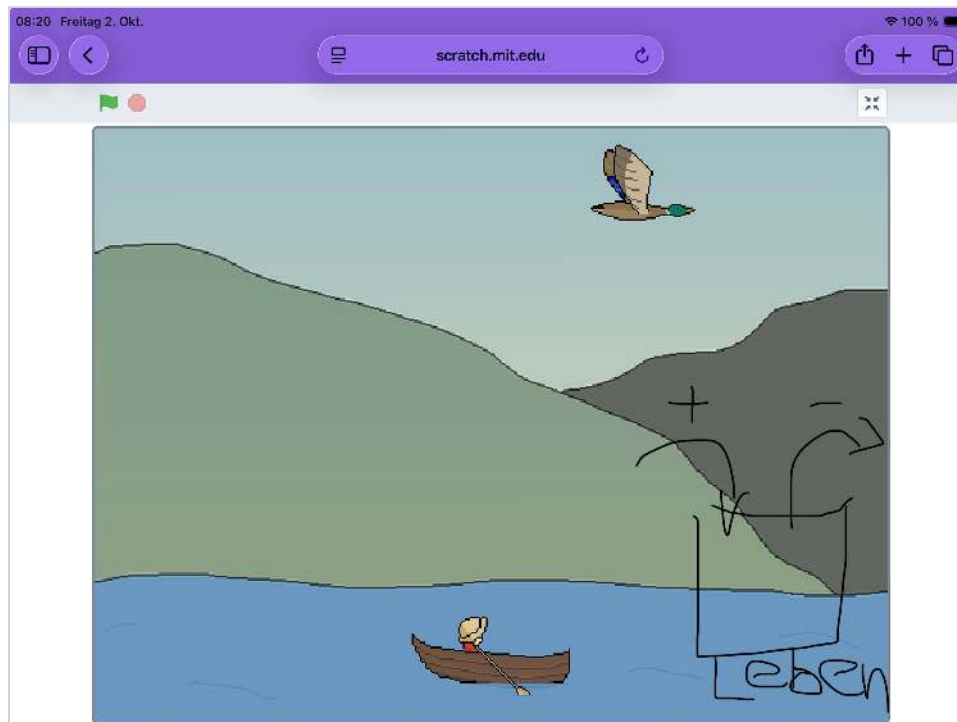
Was du in diesem Block lernst

- Eine **Variable** anlegen: ein Zähler für die Leben
- Erkennen, ob die Kacke das Boot berührt
- **Nachrichten** zwischen Figuren senden: Die Kacke meldet den Treffer, das Boot reagiert
- Das Boot blinkt
- Bei 0 Leben: Game Over

6.1 Was ist eine Variable? (3 min)

Das Spiel muss sich etwas **merken**: Wie viele Leben hat das Boot noch?

Dafür gibt es **Variablen**. Eine Variable ist wie ein **beschrifteter Becher**. Auf dem Etikett steht der Name („Leben“), im Becher liegt eine Zahl (5). Du kannst jederzeit nachsehen, was drin ist, etwas hineinlegen oder herausnehmen.



Drei Dinge machen wir mit der Variable:

Block (Kategorie Variablen , dunkelorange)	Was er tut
„setze (Leben) auf (5)“	Becher leeren, 5 hineinlegen. Am Spielstart.
„ändere (Leben) um (-1)“	Eins herausnehmen. Bei jedem Treffer.
„(Leben)“ (runder Block)	Nachsehen, was drin ist. Zum Vergleichen.

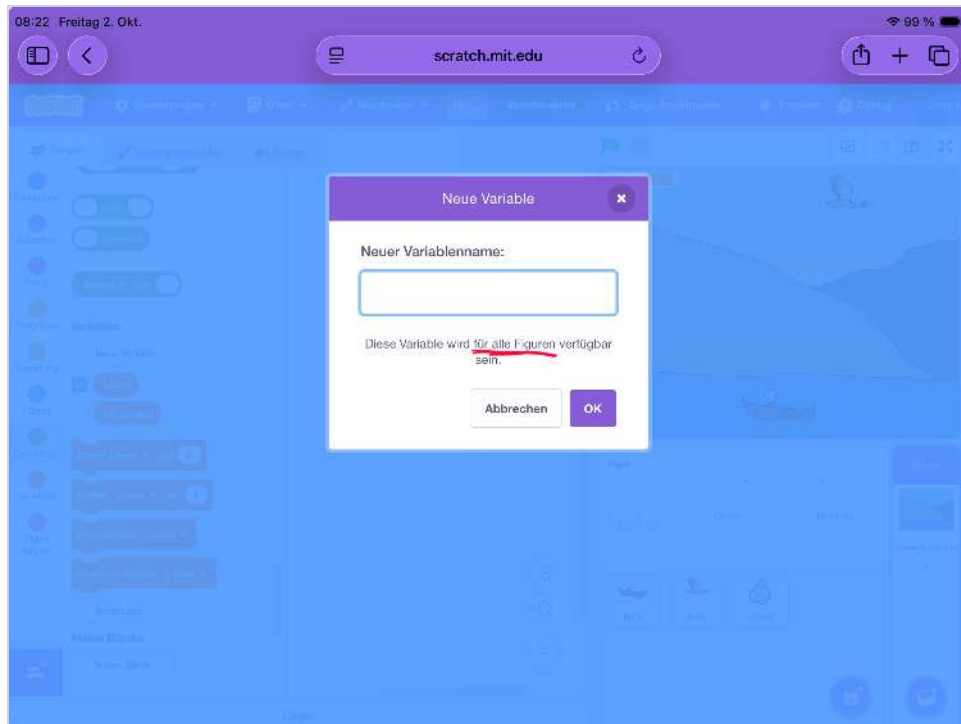
6.2 Übung: Variable „Leben“ anlegen (4 min)

Wähle in der Figurenliste rechts neben den Figuren die **Bühne**. Die Bühne ist der Spielleiter: Sie zählt Leben und beendet das Spiel.

Schritt 1: Variable erstellen

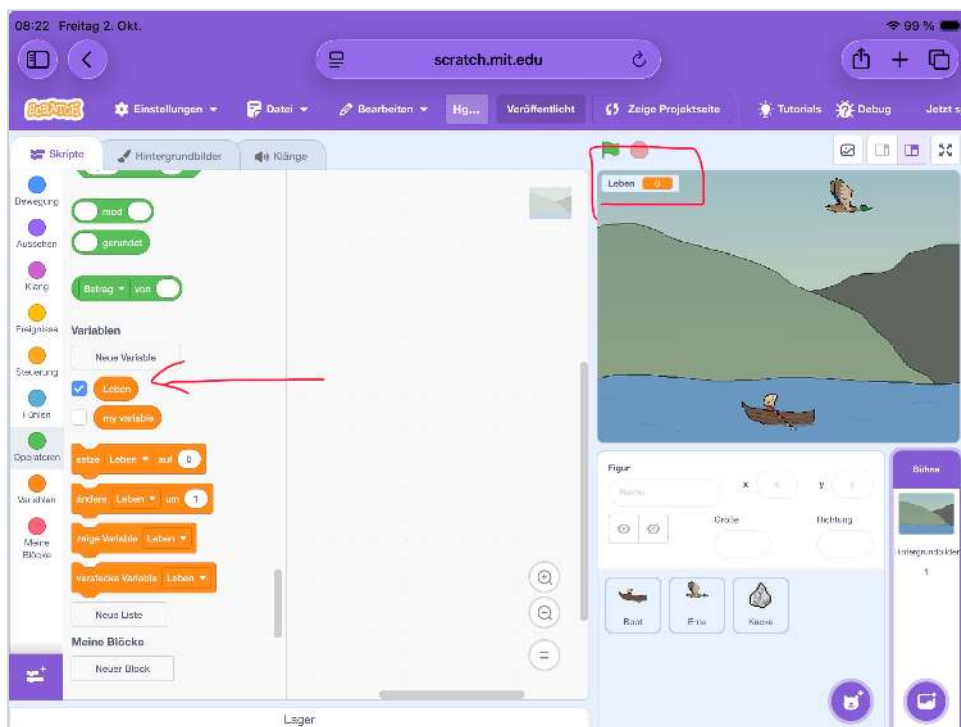
1. Kategorie **Variablen** (dunkelorange).
2. Tippe auf den Knopf „**Neue Variable**“.
3. Name: Leben
4. „**Für alle Figuren**“ muss ausgewählt sein (ist Standard bei der Bühne).
5. **OK**.

Jetzt gibt es in der Palette neue Blöcke mit „Leben“ drin.



Schritt 2: Anzeige auf der Bühne

Neben dem runden Block „(Leben)“ in der Palette ist ein **Häkchen-Kästchen**. Ist es angehakt, wird die Variable oben links auf der Bühne angezeigt. Lass es an.

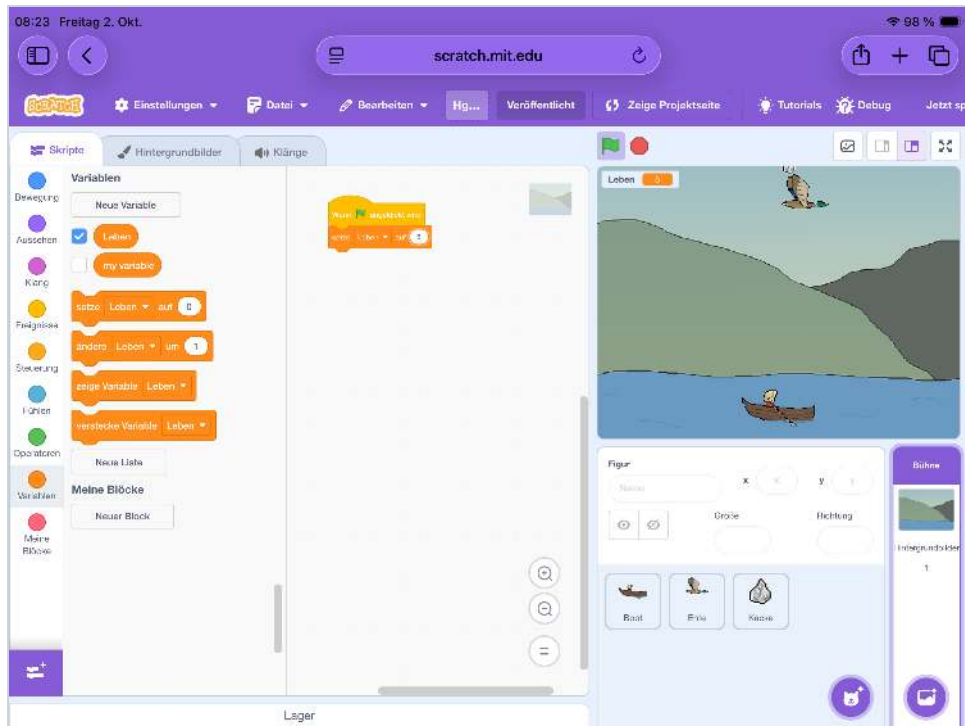


Schritt 3: Startwert setzen

1. **Ereignisse** → „Wenn  angeklickt wird“.
2. **Variablen** → „setze (Leben) auf (0)“ darunter. Ändere 0 auf **5**.

Wenn  angeklickt wird
setze [Leben] auf (5)

Grüne Flagge: Oben links steht „Leben 5“.



Leitung: Wer die Variable versehentlich bei einer Figur mit „Nur für diese Figur“ anlegt, sieht sie bei der Bühne nicht. Lösung: Variable lange drücken → löschen, bei der Bühne neu anlegen. Deshalb Variable ausdrücklich **bei der Bühne** anlegen lassen. Zeigen, wie man die Anzeige auf der Bühne mit dem Finger an eine andere Stelle zieht.

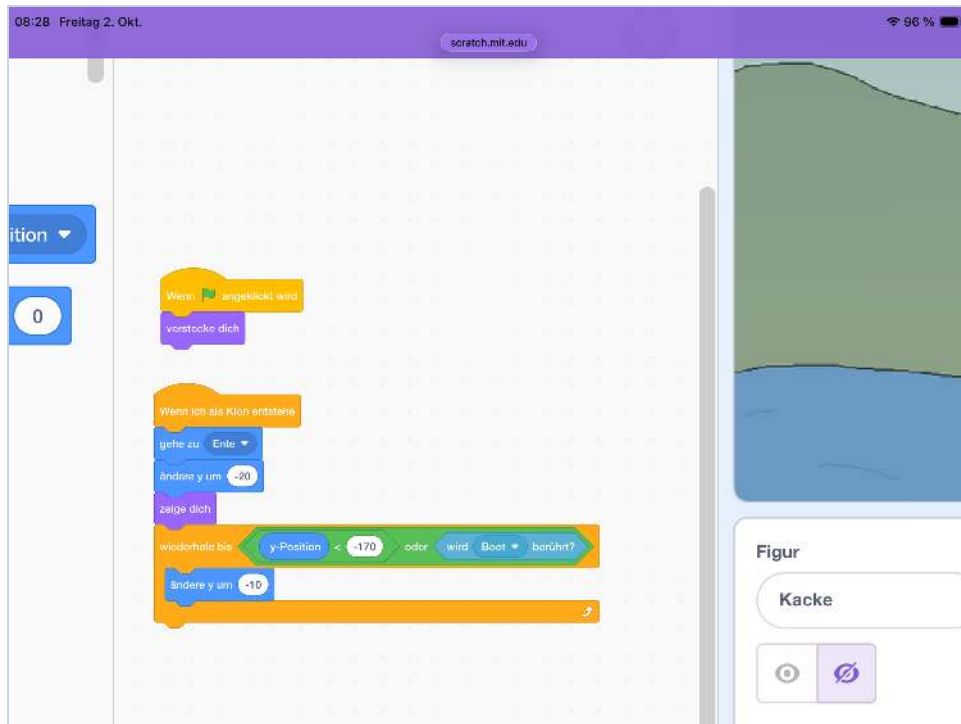
6.3 Übung: Die Kacke merkt den Treffer (5 min)

Wähle die **Kacke**. Ihr Klon-Skript aus Block 5 bekommt zwei Ergänzungen.

Schritt 1: Zweite Stopp-Bedingung

Bisher fällt die Kacke, bis sie unten ist. Jetzt soll sie **auch** aufhören, wenn sie das Boot berührt.

1. **Operatoren** (grün) → „<> oder <>“ an eine freie Stelle ziehen.
2. Ziehe den vorhandenen Block „(y-Position) < (-170)“ aus dem „wiederhole bis“ heraus und in das **linke** Loch des „oder“-Blocks.
3. **Fühlen** (hellblau) → „wird (Mauszeiger) berührt?“ in das **rechte** Loch. Dropdown → „**Boot**“.
4. Ziehe den ganzen „oder“-Block zurück in das Loch von „wiederhole bis“.



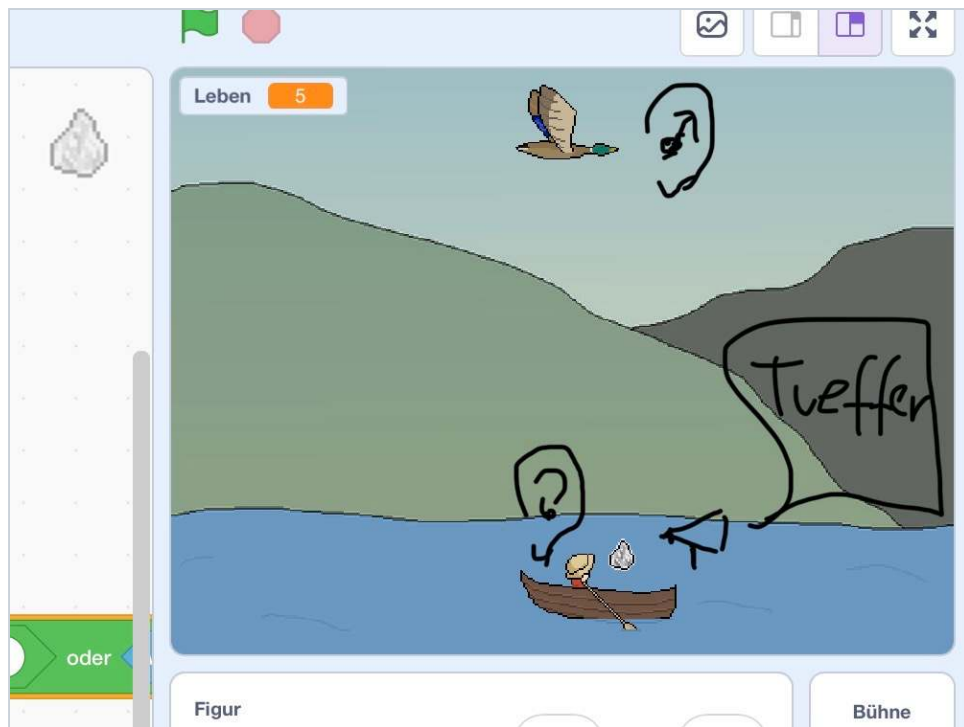
Schritt 2: Treffer melden

Nach der Schleife, **vor** „lösche diesen Klon“:

1. **Steuerung** → „falls <> dann“ zwischen Schleife und „lösche diesen Klon“ einrasten.
2. **Fühlen** → „wird (Boot) berührt?“ in das Loch von „falls“.
3. **Ereignisse** → „sende (Nachricht1) an alle“ in das „falls“ hinein.
4. Dropdown „**Nachricht1**“ antippen → „**Neue Nachricht**“ → Name: Treffer → OK.

Dein Klon-Skript:

```
Wenn ich als Klon entstehe
gehe zu [Ente]
ändere y um (-20)
zeige dich
wiederhole bis <<(y-Position) < (-170)> oder <wird [Boot] berührt?>>
  ändere y um (-5)
falls <wird [Boot] berührt?> dann
  sende [Treffer] an alle
lösche diesen Klon
```

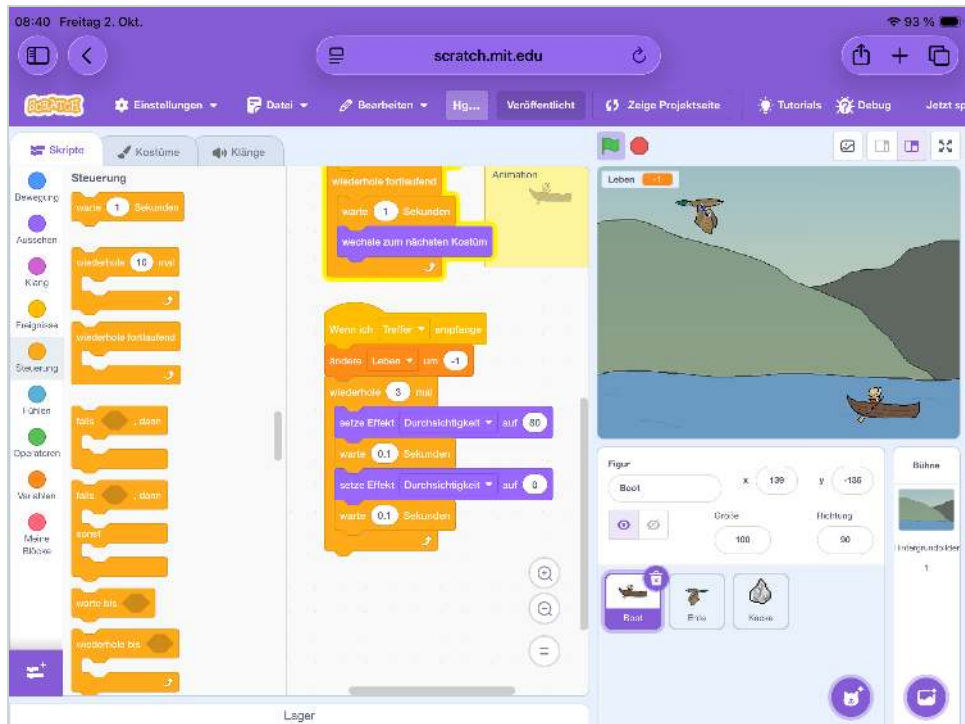
Warum nicht einfach die Kacke die Leben abziehen lassen? Ginge auch. Aber das Blinken gehört zum **Boot** — und später vielleicht noch ein Geräusch, ein Schild oder ein Wackeln. Also soll das Boot selbst entscheiden, was bei einem Treffer passiert. Die Kacke meldet nur.

6.5 Übung: Das Boot reagiert (5 min)

Wähle das **Boot**.

1. **Ereignisse** → „**Wenn ich (Treffer) empfangen**“ an eine freie Stelle.
2. **Variablen** → „**ändere (Leben) um (1)**“ darunter. Ändere 1 auf **-1**.
3. **Steuerung** → „**wiederhole (10) mal**“ darunter. Ändere 10 auf **3**.
4. **Aussehen** → „**setze Effekt (Farbe) auf (0)**“ in die Schleife. Dropdown → „**Durchsichtigkeit**“. Wert **80**.
5. **Steuerung** → „**warte (1) Sekunden**“ darunter, Wert **0.1**.
6. Noch einmal „**setze Effekt (Durchsichtigkeit) auf (0)**“. Tipp: Den ersten Effekt-Block lange drücken → „**Duplizieren**“, dann 80 auf 0 ändern.
7. Noch einmal „**warte (0.1) Sekunden**“.

```
Wenn ich [Treffer] empfangen
  ändere [Leben] um (-1)
  wiederhole (3) mal
    setze Effekt [Durchsichtigkeit] auf (80)
    warte (0.1) Sekunden
    setze Effekt [Durchsichtigkeit] auf (0)
    warte (0.1) Sekunden
```



Grüne Flagge, lass dich treffen: Das Boot blinkt dreimal, Leben 5 → 4.

Noch still? Stimmt — ein Treffer darf auch knallen. Der Klang kommt in der Kreativzeit dazu, dann mit deinem eigenen Geräusch.

Speichern!

🎓 **Leitung:** Duplizieren am Beamer mit Rechtsklick, auf dem iPad lange drücken. Beim Duplizieren hängt die Kopie am Finger, bis man sie ablegt. Klang bewusst nicht hier — kommt in Block 9 als eigene Idee, damit die Teilnehmer ihn selbst einbauen (und das Treffer-Skript noch einmal anfassen müssen). Wer Effekt „Farbe“ stehen lässt: Boot wird bunt statt durchsichtig, auch okay, aber auf Dropdown hinweisen.

6.6 Übung: Game Over (4 min)

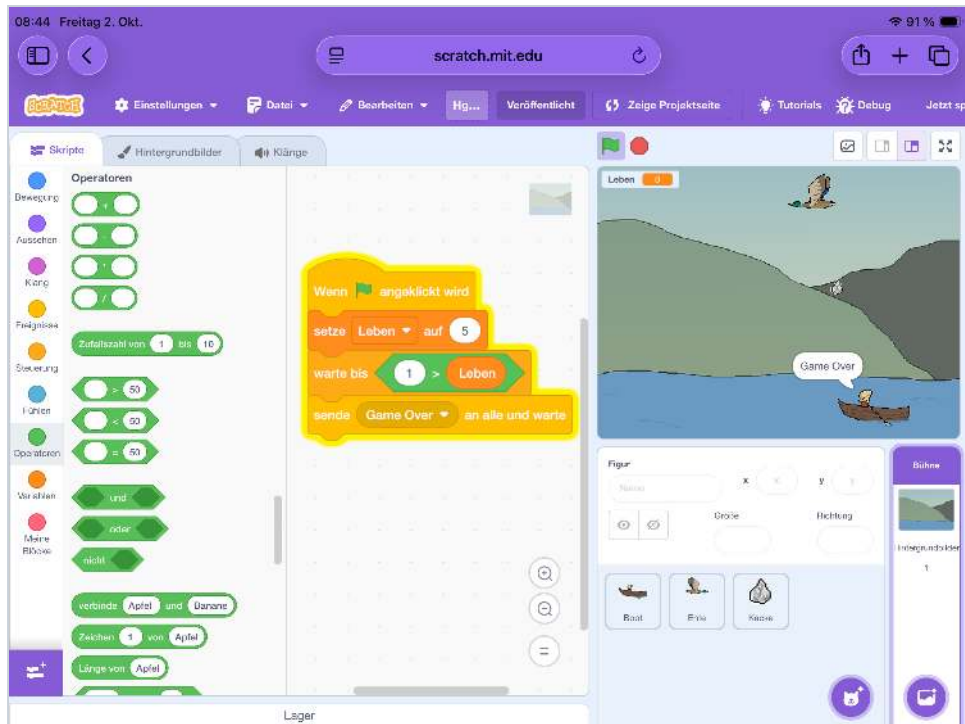
Zurück zur **Bühne**. Sie wartet, bis die Leben aufgebraucht sind.

Schritt 1: Bühne wartet und ruft

Ergänze das Skript der Bühne:

1. **Steuerung** → „**warte bis <>**“ unter „setze Leben auf 5“.
2. **Operatoren** → „**() < (50)**“ ins Loch. **Variablen** → runder Block „**(Leben)**“ ins linke Feld. Rechts **1**.
3. **Ereignisse** → „**sende (Treffer) an alle**“ darunter. Dropdown → „**Neue Nachricht**“ → Game Over.

```
Wenn [ ] angeklickt wird
  setze [Leben] auf (5)
  warte bis <(Leben) < (1)>
  sende [Game Over] an alle
```

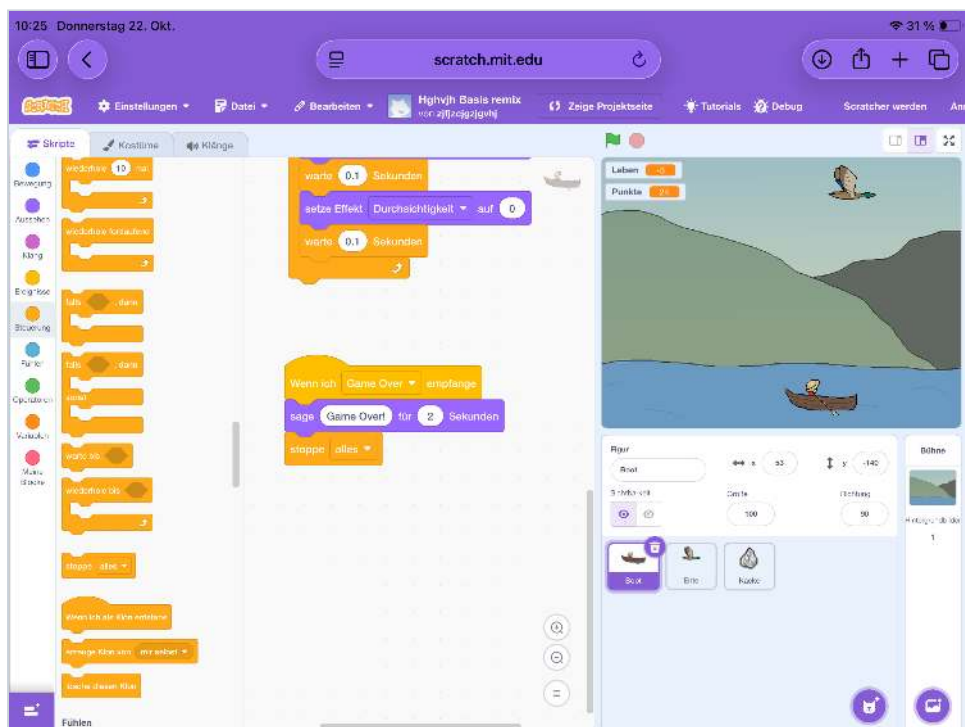


Schritt 2: Boot hört zu

Wähle das **Boot**:

1. **Ereignisse** → „Wenn ich (Game Over) empfangen“.
2. **Aussehen** → „sage (Hallo!) für (2) Sekunden“, Text Game Over!.
3. **Steuerung** → „stoppe (alle)“ darunter.

Wenn ich [Game Over] empfangen
sage (Game Over!) für (2) Sekunden
stoppe [alle]

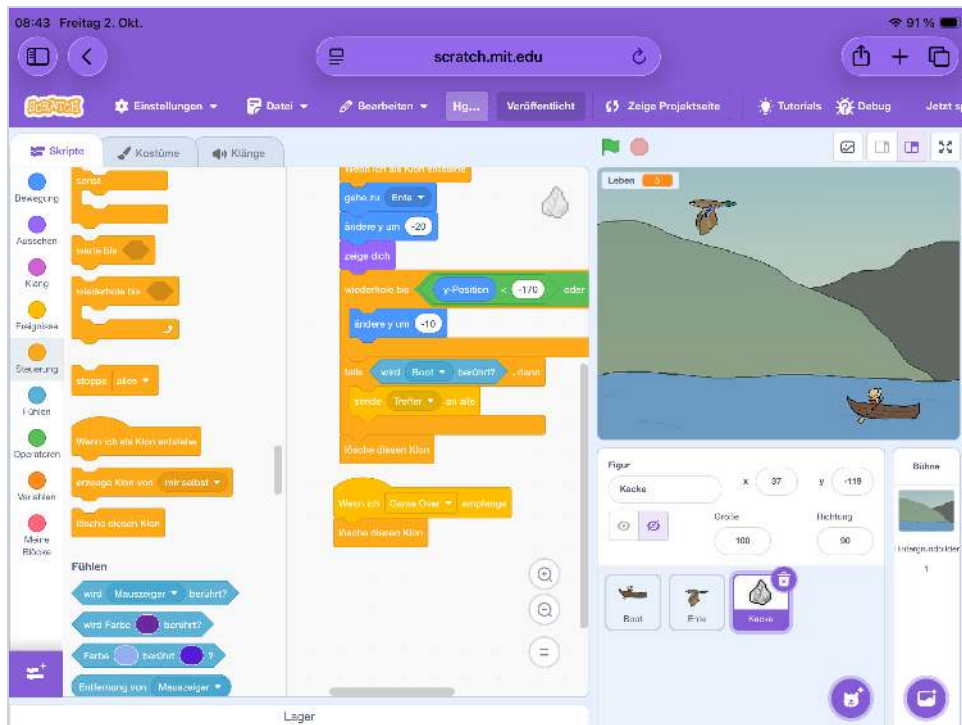


Schritt 3: Kacke hört zu

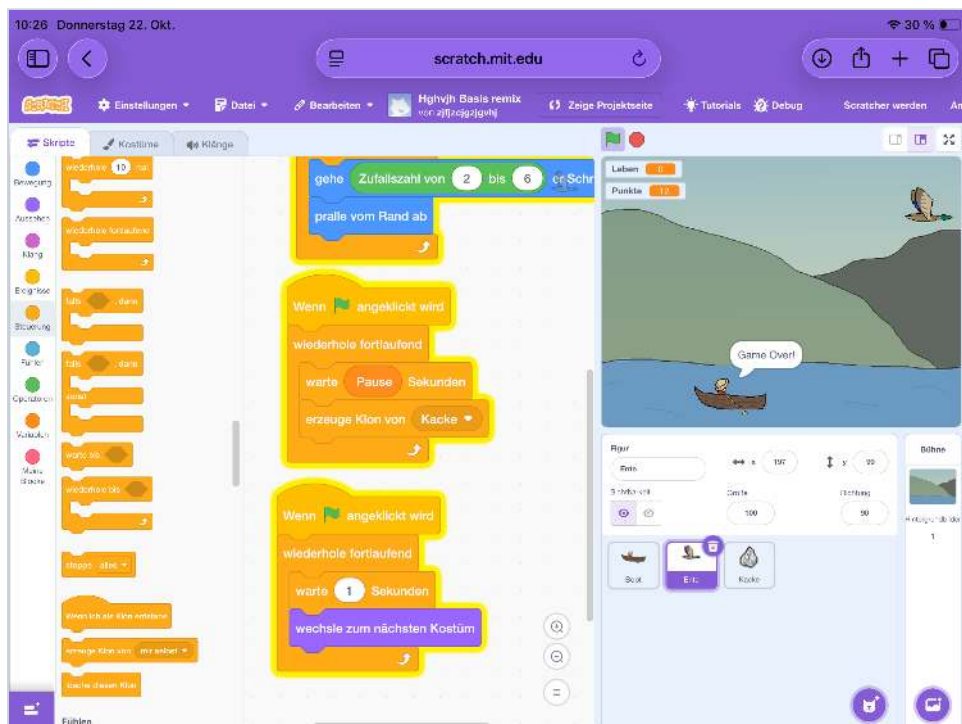
Wähle die **Kacke**:

1. **Ereignisse** → „Wenn ich (Game Over) empfangen“.
2. **Steuerung** → „lösche diesen Klon“ darunter.

Wenn ich [Game Over] empfange
lösche diesen Klon



So verschwinden alle Kacken, die noch in der Luft sind.



Grüne Flagge, lass dich fünfmal treffen. „Game Over!“, alles stoppt.

Speichern!

6.7 Was ist passiert? (2 min)

- **Variable** „Leben“: ein Becher, in den alle Figuren schauen dürfen („Für alle Figuren“). Die Bühne füllt ihn auf 5, das Boot nimmt bei jedem Treffer 1 raus, die Bühne prüft ständig, ob er leer ist.
- **„oder“**: Die Kacke hört auf zu fallen, wenn sie unten ist **oder** das Boot berührt. Eine der beiden Antworten muss wahr sein.
- **Nachrichten** koppeln Figuren locker: Die Kacke weiß nichts über Klänge oder Blinken. Sie ruft nur. Das Boot regelt den Rest. Wenn du später ein Schild „Autsch!“ einblenden willst, hört einfach eine

weitere Figur auf „Treffer“.

- **„warte bis“** blockiert das Skript, bis die Bedingung wahr ist. Die Bühne steht also die ganze Runde bei diesem Block und wartet.
- **„stoppe alle“** beendet alle Skripte, wie das rote Stoppschild.

Herausforderung (wenn du schneller bist)

Such dir eine aus:

1. **Autsch:** Das Boot sagt beim Treffer kurz „Autsch!“ (Aussehen → „sage ... für 0.5 Sekunden“ im Treffer-Skript).
2. **Ente lacht:** Die Ente hört auch auf „Treffer“ und wechselt kurz die Größe: „ändere Größe um 20“, warte 0.2, „ändere Größe um -20“.
3. **Kleiner werden:** Statt zu blinken, schrumpft das Boot pro Treffer: „ändere Größe um -10“. Am Anfang „setze Größe auf 100 %“ nicht vergessen.
4. **Härter:** Starte mit 3 Leben statt 5. Oder: Jede Kacke zieht 2 Leben ab. Was passiert dann mit „Leben < 1“?

Checkliste Block 6

- ☐ Die Variable „Leben“ ist bei der Bühne angelegt, „Für alle Figuren“, und wird auf der Bühne angezeigt.
- ☐ Am Start steht Leben auf 5.
- ☐ Trifft eine Kacke das Boot: Blinken, Leben -1.
- ☐ Bei 0 Leben sagt das Boot „Game Over!“ und alles stoppt.
- ☐ Ich habe auf „Jetzt speichern“ getippt.

Weiter geht's: In Block 7 wird das Spiel mit der Zeit schwerer: Punkte zählen, Kacke fällt immer öfter.

Leitung – Ablauf Block 6 (25 min)

Min	Was
0–3	Becher-Bild für Variablen, drei Variablen-Blöcke zeigen
3–7	Übung 6.2 Variable anlegen bei der Bühne, Anzeige, Startwert
7–12	Übung 6.3 Kacke: „oder“-Bedingung, „falls berührt Boot → sende Treffer“, neue Nachricht anlegen
12–14	Lautsprecher-Bild für Nachrichten
14–19	Übung 6.5 Boot reagiert: Leben -1, Blinken
19–23	Übung 6.6 Game Over: Bühne wartet, Boot und Kacke hören zu
23–25	„Was ist passiert?“, speichern, Pause ansagen

Vorbereiten: Zwischenstand „Entenspiel Stand 05 – Es kackt“ im Klassenstudio. Vor der Pause: alle speichern lassen, Orga-Betreuer prüft am Studio, wessen Projekt zuletzt gespeichert wurde.

Beamer über Laptop: Duplizieren = Rechtsklick am Laptop, lange drücken auf dem iPad. Block aus einem Loch herausziehen: genau den inneren Block anfassen. Häkchen für die Bühnenanzeige ist ein kleines Kästchen, auf dem iPad gut zielen.

Typische Probleme: Variable „Nur für diese Figur“ angelegt → Bühne sieht sie nicht · „ändere Leben um 1“ statt -1 (Leben steigt) · Nachricht „Treffer“ bei Kacke und Boot unterschiedlich geschrieben (zwei Nachrichten „Treffer“ und „treffer“) · „Wenn ich empfangen“ beim falschen Empfänger · „lösche diesen Klon“ fehlt nach dem „falls“ → Kacke bleibt am Boot kleben und zählt weiter runter · „warte bis Leben < 1“ mit „=“ gebaut, funktioniert auch, aber nicht bei Herausforderung 4.

Block 7: Schwieriger mit der Zeit (20 min)

Teilnehmer-Handout. Kästen mit  sind Hinweise für die Workshop-Leitung und werden vor dem Druck entfernt. Platzhalter für Bilder:  [Bild: ...]

Was du in diesem Block lernst

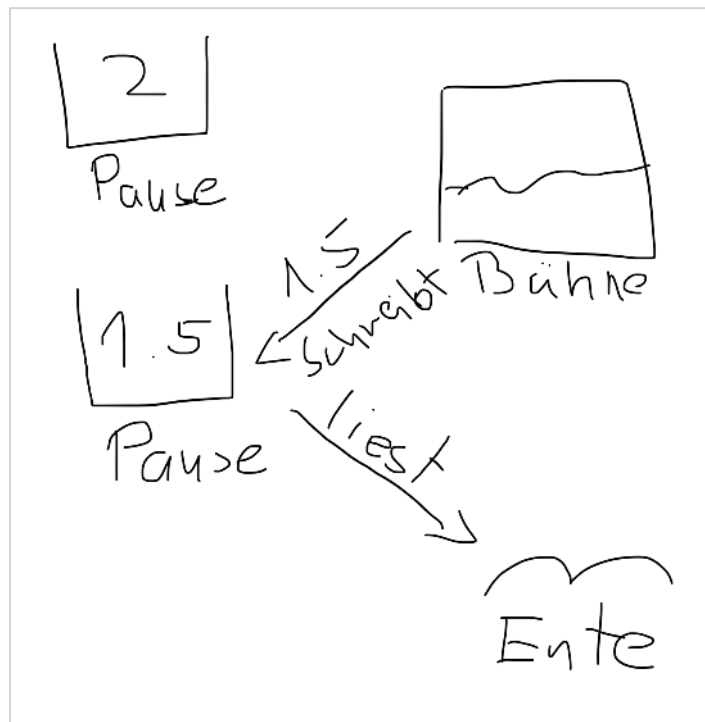
- Punkte zählen: Jede Sekunde, die du überlebst, gibt einen Punkt
- Eine Variable als **Stellschraube** benutzen: Die Wartezeit der Ente wird eine Variable
- Mit Blöcken **rechnen** und vergleichen
- Das Spiel wird von sich aus immer schwerer, mit einer Untergrenze

7.1 Warum Variablen als Stellschraube? (2 min)

Bis jetzt kackt die Ente **alle 2 Sekunden**. Die 2 steht fest im Block. Wenn du sie ändern willst, musst du in das Skript der Ente gehen.

Was, wenn die Zahl sich **während des Spiels** ändern soll? Am Anfang 2 Sekunden, nach einer Minute nur noch 1 Sekunde, später 0.5?

Dafür kommt die Zahl in eine Variable: **„Pause“**. Die Ente liest bei jedem Durchgang nach, was gerade im Becher „Pause“ liegt. Die Bühne dreht nebenbei an der Stellschraube und macht die Pause immer kleiner.

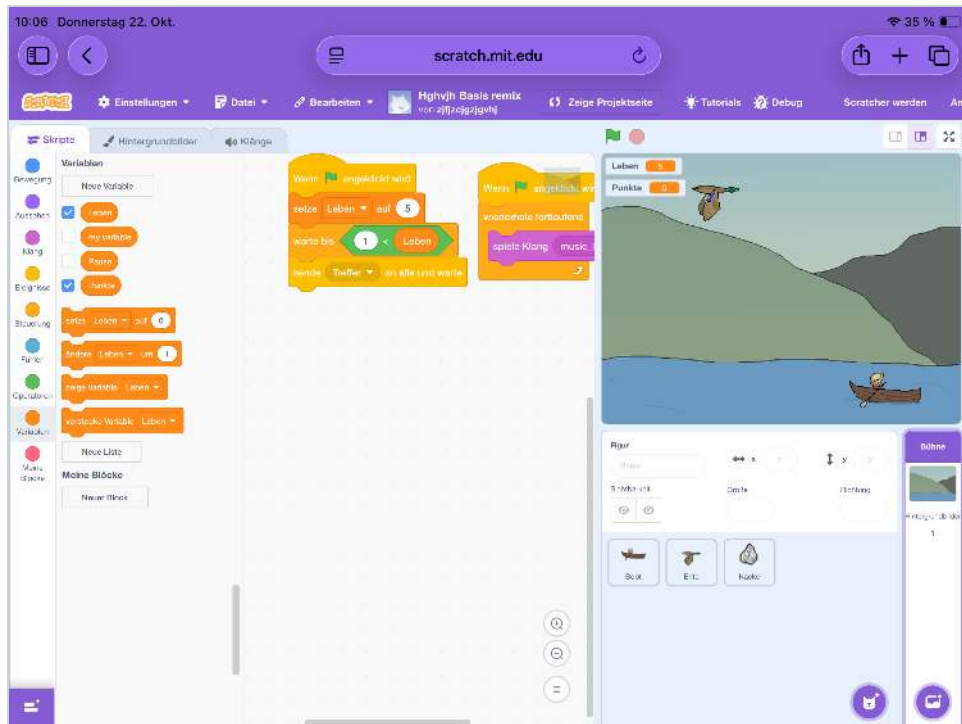


7.2 Übung: Punkte zählen (5 min)

Wähle die **Bühne**. Neben dem Leben-Skript bauen wir ein **zweites** Skript.

Schritt 1: Zwei neue Variablen

1. **Variablen** → „**Neue Variable**“ → Name Punkte → Für alle Figuren → OK.
2. Noch einmal „**Neue Variable**“ → Name Pause → Für alle Figuren → OK.
3. Häkchen: **Punkte** angehakt (soll auf der Bühne stehen). **Pause** Häkchen **weg** (die Spieler müssen sie nicht sehen).



Schritt 2: Startwerte

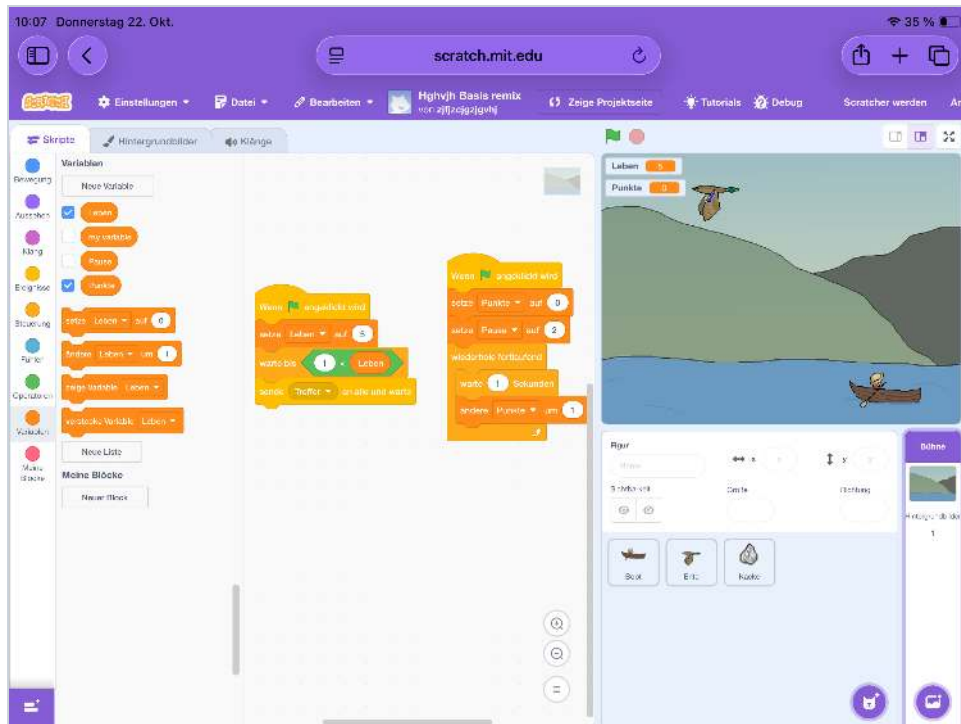
1. **Ereignisse** → „**Wenn** **angeklickt wird**“ an eine freie Stelle.
2. **Variablen** → „**setze (Punkte) auf (0)**“ darunter. (Dropdown auf Punkte, falls eine andere Variable drinsteht.)
3. **Variablen** → „**setze (Pause) auf (2)**“ darunter.

Schritt 3: Jede Sekunde ein Punkt

1. **Steuerung** → „**wiederhole fortlaufend**“ darunter.
2. **Steuerung** → „**warte (1) Sekunden**“ in die Schleife.
3. **Variablen** → „**ändere (Punkte) um (1)**“ darunter in die Schleife.

```

Wenn  angeklickt wird
  setze [Punkte] auf (0)
  setze [Pause] auf (2)
  wiederhole fortlaufend
    warte (1) Sekunden
    ändere [Punkte] um (1)
  
```



Grüne Flagge: Oben links zählen die Punkte hoch. Bei Game Over bleiben sie stehen („stoppe alle“ stoppt auch dieses Skript).

Leitung: Dropdown im „setze“-Block zeigt die zuletzt angelegte Variable, oft steht „Pause“ drin, wo „Punkte“ hin soll. Kontrollieren lassen. Die Anzeigen auf der Bühne überlappen manchmal: mit dem Finger auseinanderziehen.

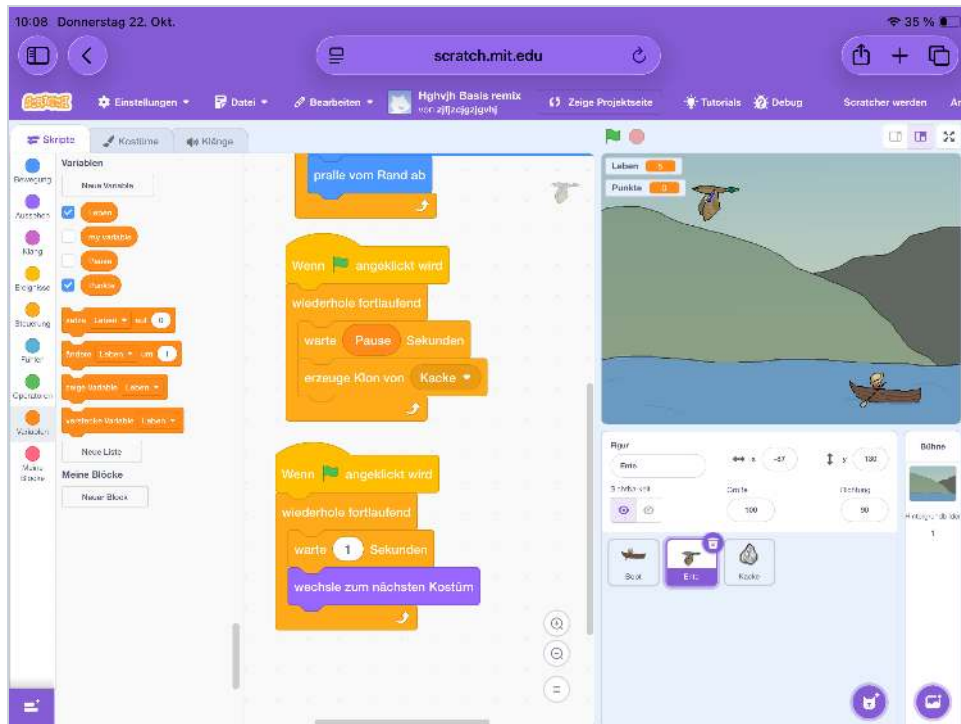
7.3 Übung: Die Stellschraube einbauen (4 min)

Schritt 1: Die Ente liest die Pause

Wähle die **Ente**. Im Stempel-Skript aus Block 5 steht warte (2) Sekunden.

1. **Variablen** → runder Block **„(Pause)“** in das Zahlenfeld des warte-Blocks ziehen. Die 2 verschwindet, der runde Block sitzt im Loch.

Wenn angeklickt wird
wiederhole fortlaufend
warte (Pause) Sekunden
erzeuge Klon von [Kacke]



Grüne Flagge: Genau wie vorher, alle 2 Sekunden. Die Bühne setzt Pause ja auf 2. Der Unterschied kommt jetzt.

Schritt 2: Die Bühne dreht an der Schraube

Wähle die **Bühne**. Im Punkte-Skript, in der Schleife, unter „ändere Punkte um 1“:

1. **Steuerung** → „falls <> dann“.
2. **Operatoren** → „() > (50)“ ins Loch. **Variablen** → runder Block „(Pause)“ ins linke Feld. Rechts **0.4**.
3. **Variablen** → „ändere (Pause) um (1)“ in das „falls“. Wert **-0.05**.

```
Wenn [ ] angeklickt wird
  setze [Punkte] auf (0)
  setze [Pause] auf (2)
  wiederhole fortlaufend
    warte (1) Sekunden
    ändere [Punkte] um (1)
    falls <(Pause) > (0.4)> dann
      ändere [Pause] um (-0.05)
```

[Bild: Fertiges Punkte-Skript der Bühne mit „falls Pause > 0.4“ — assets/07_punkte-skript.PNG neu aufnehmen, die vorhandene Aufnahme zeigt im „ändere“-Block „my variable“ statt „Pause“]

Grüne Flagge und durchhalten: Nach 20 Sekunden ist die Pause bei 1 Sekunde, nach 30 Sekunden bei 0.5. Es regnet Kacke.

Speichern!

Leitung: Zum Testen die Anzeige von „Pause“ kurz einschalten, damit alle sehen, wie die Zahl schrumpft. Danach wieder aus. Wer den Effekt schneller sehen will: -0.05 vorübergehend auf -0.2.

7.4 Was ist passiert? (3 min)

Variable statt fester Zahl

Der runde Block „(Pause)“ passt in jedes Zahlenfeld. Scratch schaut dann **jedes Mal neu** nach, was gerade drin ist. Die Ente merkt gar nicht, dass sich etwas ändert. Sie macht einfach „warte (Pause)“.

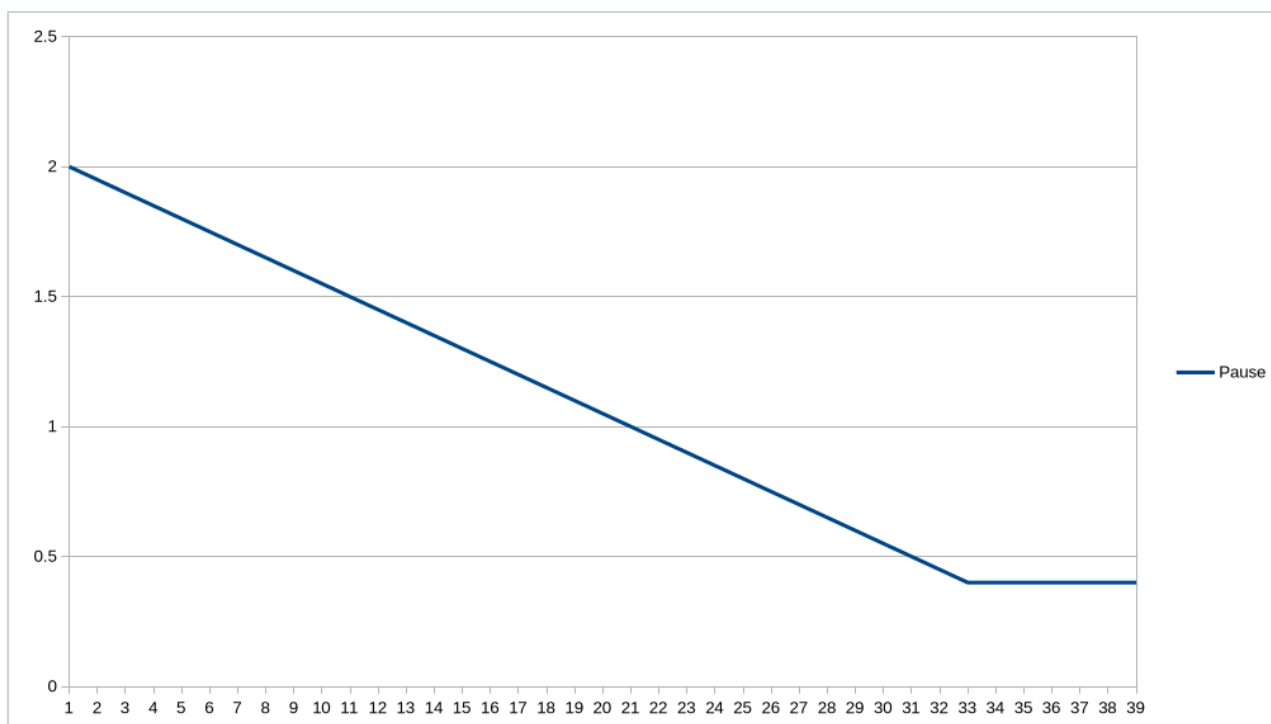
Rechnen mit Blöcken

ändere Pause um -0.05 heißt: $\text{Pause} = \text{Pause} - 0.05$. Jede Sekunde ein bisschen weniger. Nach 20 Sekunden: $2 - 20 \times 0.05 = 1$.

Warum die Untergrenze „> 0.4“?

Ohne das „falls“ würde die Pause immer weiter sinken: 0.1, 0, dann **negativ**. Bei „warte –1 Sekunden“ wartet Scratch gar nicht mehr. Die Ente würde in jedem einzelnen Bild eine Kacke stempeln, hunderte pro Sekunde. Das Spiel wird unspielbar und das iPad langsam.

Also: Nur kleiner machen, **solange** die Pause noch über 0.4 liegt. Unter 0.4 bleibt sie stehen. 0.4 ist die maximale Schwierigkeit. Du darfst die Zahl ändern, aber probiere vorher aus, was noch schaffbar ist.



Die Bühne als Spielleiter

Die Bühne hat jetzt zwei Skripte: Eines zählt Leben und wartet auf Game Over, eines zählt Punkte und dreht an der Schwierigkeit. Die Figuren kümmern sich nur um sich selbst. Das ist eine gute Aufteilung.

Herausforderung (wenn du schneller bist)

A: Die Ente wird schneller (★★)

Noch eine Stellschraube, diesmal für die Fluggeschwindigkeit der Ente.

1. **Bühne:** Neue Variable Tempo, Für alle Figuren, Häkchen aus.
2. Im Punkte-Skript oben: „**setze (Tempo) auf (4)**“.
3. In der Schleife: „**ändere (Tempo) um (0.1)**“.
4. **Ente:** Im Flug-Skript aus Block 4 den runden Block „**(Tempo)**“ in das Feld von „**gehe (4)er Schritt**“ ziehen.

```
Ente:  
wiederhole fortlaufend  
  gehe (Tempo)er Schritt  
  pralle vom Rand ab
```

Braucht das Tempo auch eine Obergrenze? Probiere aus, was bei Tempo 20 passiert.

B: Eine zweite Ente nach 30 Sekunden (★★★★)

Dafür brauchen wir eine zweite Enten-**Figur** und eine zweite Kacke-Figur. Warum nicht einfach einen Klon der Ente? Weil der Kacke-Klon mit „gehe zu (Ente)“ immer zum **Original** springt. Eine zweite Ente würde ihre Kacke also immer bei der ersten Ente fallen lassen.

Schritt 1: Ente duplizieren

1. In der Figurenliste die **Ente lange drücken** → „**Duplizieren**“. Es entsteht **Ente2** mit allen drei Skripten.

Schritt 2: Kacke duplizieren

1. **Kacke** lange drücken → „**Duplizieren**“ → **Kacke2**.
2. Bei **Kacke2** im Klon-Skript: Dropdown von „**gehe zu (Ente)**“ auf „**Ente2**“.

Schritt 3: Ente2 anpassen

1. Im Stempel-Skript: Dropdown „**erzeuge Klon von (Kacke)**“ auf „**Kacke2**“.
2. Im Flug-Skript: gehe zu x: (-200) y: (130) ändern auf **x: 200**, damit sie rechts startet.
3. Im Flug-Skript direkt unter dem Hut-Block einfügen: „**verstecke dich**“, „**warte (30) Sekunden**“, „**zeige dich**“.
4. Im Stempel-Skript vor der Schleife ebenfalls „**warte (30) Sekunden**“, sonst kackt die unsichtbare Ente schon.

```
Ente2 – Flug:  
Wenn  angeklickt wird  
verstecke dich  
warte (30) Sekunden  
zeige dich  
gehe zu x: (200) y: (130)  
setze Drehtyp auf [links-rechts]  
setze Richtung auf (90)  
wiederhole fortlaufend  
  gehe (4)er Schritt  
  pralle vom Rand ab
```

```
Ente2 – Stempel:  
Wenn  angeklickt wird  
warte (30) Sekunden  
wiederhole fortlaufend  
  warte (Pause) Sekunden  
  erzeuge Klon von [Kacke2]
```

Vergiss nicht: Auch **Kacke2** braucht „Wenn ich (Game Over) empfangen → lösche diesen Klon“. Das hat sie durch das Duplizieren schon mitbekommen. Prüfe es.

C: Bonus für Mutige (★)

Ändere die Untergrenze 0.4 auf 0.2 und spiele 60 Sekunden. Notiere deinen Punktestand. Ist das noch fair?

Checkliste Block 7

- ☐ Die Variable „Punkte“ zählt jede Sekunde hoch und wird angezeigt.
- ☐ Die Variable „Pause“ steuert die Wartezeit der Ente.
- ☐ Die Pause wird jede Sekunde kleiner, aber nie kleiner als 0.4.
- ☐ Ich kann erklären, warum die Untergrenze wichtig ist.
- ☐ Ich habe auf „Jetzt speichern“ getippt.

Weiter geht's: In Block 8 kommt der Feinschliff: Musik, Punkte-Sound, Game-Over-Anzeige mit Punktestand.

🎓 Leitung – Ablauf Block 7 (20 min)

Min	Was
0–2	Stellschrauben-Bild: feste Zahl vs. Variable
2–7	Übung 7.2 Punkte zählen, zwei neue Variablen, Häkchen
7–11	Übung 7.3 „Pause“ in die Ente, Bühne verkleinert Pause mit „falls > 0.4“
11–14	„Was ist passiert?": Rechnen, Untergrenze, negative Wartezeit vorführen
14–19	Herausforderungen A/B/C, Orga-Betreuer prüft Checkliste
19–20	Speichern

Vorbereiten: Zwischenstand „Entenspiel Stand 06 – Treffer & Leben“ im Klassenstudio. Eigenes Projekt mit Herausforderung B fertig, um sie zeigen zu können, falls jemand dort landet.

Beamer über Laptop: Figur duplizieren = Rechtsklick auf die Figur in der Figurenliste, auf dem iPad lange drücken. Runden Variablen-Block in ein Zahlenfeld ziehen: Zielen auf das weiße Feld, bis es weiß aufleuchtet, dann loslassen. Auf dem iPad den Skriptbereich vorher mit „=" zurücksetzen, damit Felder groß genug sind.

Negative Wartezeit vorführen: Untergrenze kurz entfernen, Pause auf 0 setzen, Flagge. Kacke-Sturzflut, dann Stoppschild. Bleibt hängen: Grenzen sind wichtig.

Typische Probleme: falsches Dropdown in „setze“/„ändere“ (Pause statt Punkte) · runder „Pause“-Block liegt neben dem warte-Block statt im Feld · „>“ und „<“ vertauscht (Pause wird nie kleiner) · „ändere Pause um 0.05“ ohne Minus (Spiel wird leichter) · Herausforderung B: „gehe zu Ente“ bei Kacke2 nicht umgestellt · Ente2 kackt schon während sie versteckt ist (warte 30 im Stempel-Skript vergessen).

Block 8: Feinschliff (15 min)

Teilnehmer-Handout. Kästen mit 🎓 sind Hinweise für die Workshop-Leitung und werden vor dem Druck entfernt.

Was du in diesem Block lernst

- Musik, die im Hintergrund durchläuft
- Ein Geräusch, das alle 10 Punkte kommt (mit dem Rechen-Block „mod“)
- Text und Zahl zu einem Satz zusammenkleben („verknüpfen“)
- Das Spiel sauber zurücksetzen, wenn es neu startet

Dein Spiel funktioniert schon. Jetzt machen wir es rund: Musik, Belohnungs-Sound, eine richtige Game-Over-Anzeige.

8.1 Hintergrundmusik (3 min)

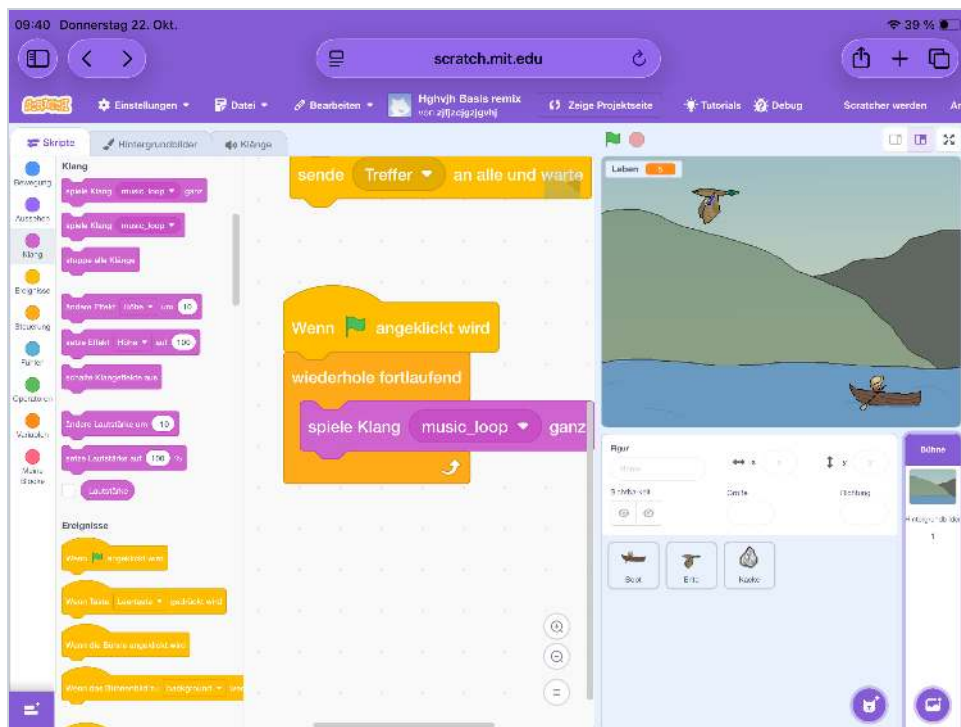
Musik gehört zur ganzen Bühne, nicht zu einer Figur. Tippe deshalb rechts unten in der Figurenliste auf die **Bühne**.

Die Bühne hat den Klang **music_loop** zugeordnet. Du kannst das im Tab „**Klänge**“ prüfen.

Baue auf der Bühne ein neues Skript:

```
Wenn 🚩 angeklickt wird  
wiederhole fortlaufend  
spiele Klang [music_loop] ganz
```

- **Ereignisse** (gelb): „Wenn 🚩 angeklickt wird“
- **Steuerung** (orange): „wiederhole fortlaufend“
- **Klang** (pink): „spiele Klang (music_loop) ganz“



Teste mit der grünen Flagge. Die Musik läuft in Dauerschleife, weil „ganz“ wartet, bis der Klang fertig ist, und die Schleife ihn dann neu startet.

Leitung: Auf iPads Lautstärke vorher leise stellen oder Kopfhörer. 10 iPads mit Musik-Loop im selben Raum nerven schnell. Wer „spiele Klang“ statt „spiele Klang ganz“ nimmt, bekommt ein Knatzen: Der Klang wird jeden Frame neu gestartet. Kurz am Beamer vorführen, guter Aha-Moment.

8.2 Belohnungs-Sound alle 10 Punkte (4 min)

Bleib auf der **Bühne**. Du hast dort seit Block 7 das Punkte-Skript. Wir bauen darin einen Sound ein, der bei 10, 20, 30, ... Punkten kommt.

Dafür brauchst du den Block „**() mod ()**“ aus **Operatoren** (grün). „mod“ heißt: der Rest beim Teilen.

Rechnung	Ergebnis
7 mod 10	7
10 mod 10	0
23 mod 10	3
30 mod 10	0

Immer wenn der Rest **0** ist, sind die Punkte durch 10 teilbar. Genau dann spielen wir den Sound.

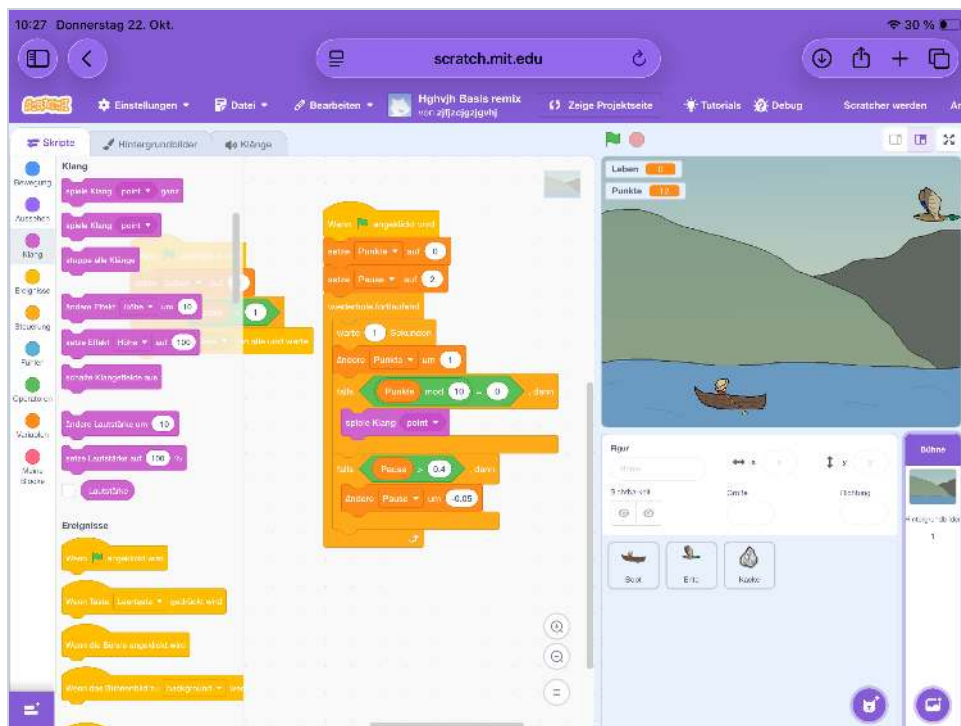
Ergänze dein Punkte-Skript direkt nach „ändere Punkte um 1“:

```
Wenn  angeklickt wird
  setze [Punkte] auf (0)
  setze [Pause] auf (2)
  wiederhole fortlaufend
    warte (1) Sekunden
    ändere [Punkte] um (1)
    falls <((Punkte) mod (10)) = (0)> dann
      spiele Klang [point]
    falls <(Pause) > (0.4)> dann
      ändere [Pause] um (-0.05)
```

So baust du die Bedingung zusammen:

1. **Steuerung** → „falls < > dann“ unter „ändere Punkte um 1“ einrasten.
2. **Operatoren** → „() = ()“ in das sechseckige Loch ziehen.

3. **Operatoren** → „() mod ()“ in das **linke** Feld des „=-“-Blocks ziehen.
4. **Variablen** → „Punkte“ in das linke Feld von „mod“ ziehen. Rechts **10** eintippen.
5. Rechtes Feld von „=-“: **0** eintippen.
6. **Klang** → „spiele Klang (point)“ in das „falls“ legen. Hier **ohne** „ganz“, sonst stockt die Schleife.



Teste. Nach 10 Sekunden kommt der erste „Pling“.

Leitung: Ineinandergeschachtelte Operatoren sind auf dem iPad fummelig. Tipp: Den „mod“-Block erst im leeren Skriptbereich fertig bauen, dann als Ganzes in den „=-“-Block ziehen, dann das Ganze ins „falls“. Vorher am Beamer zeigen, dass der Klang „point“ bei der Bühne liegt, nicht beim Boot.

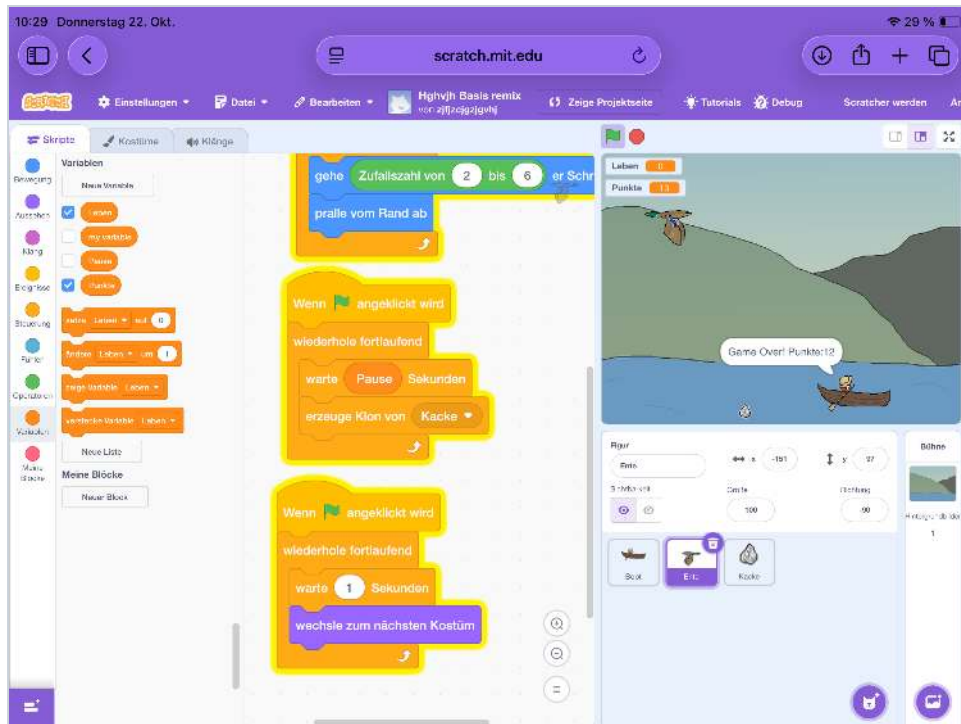
8.3 Game Over mit Punktzahl (3 min)

Beim Game Over soll das Boot sagen, wie viele Punkte du hattest. Dafür kleben wir Text und Zahl zusammen: **„verknüpfe () und ()“** aus **Operatoren**.

Tippe auf das **Boot**. Suche das Skript „Wenn ich [Game Over] empfangen“ aus Block 6. Ändere den „sage“-Block:

Wenn ich [Game Over] empfangen
sage (verknüpfe (Game Over! Punkte:) und (Punkte)) für (3) Sekunden
stoppe [alle]

1. **Operatoren** → „verknüpfe (apple) und (banana)“ in das Textfeld des „sage“-Blocks ziehen.
2. Links den Text ändern: Game Over! Punkte: (mit Leerzeichen am Ende).
3. **Variablen** → „Punkte“ in das rechte Feld ziehen.
4. Zeit auf **3** Sekunden.



Teste: Lass dich fünfmal treffen. Das Boot zeigt deine Punktzahl.

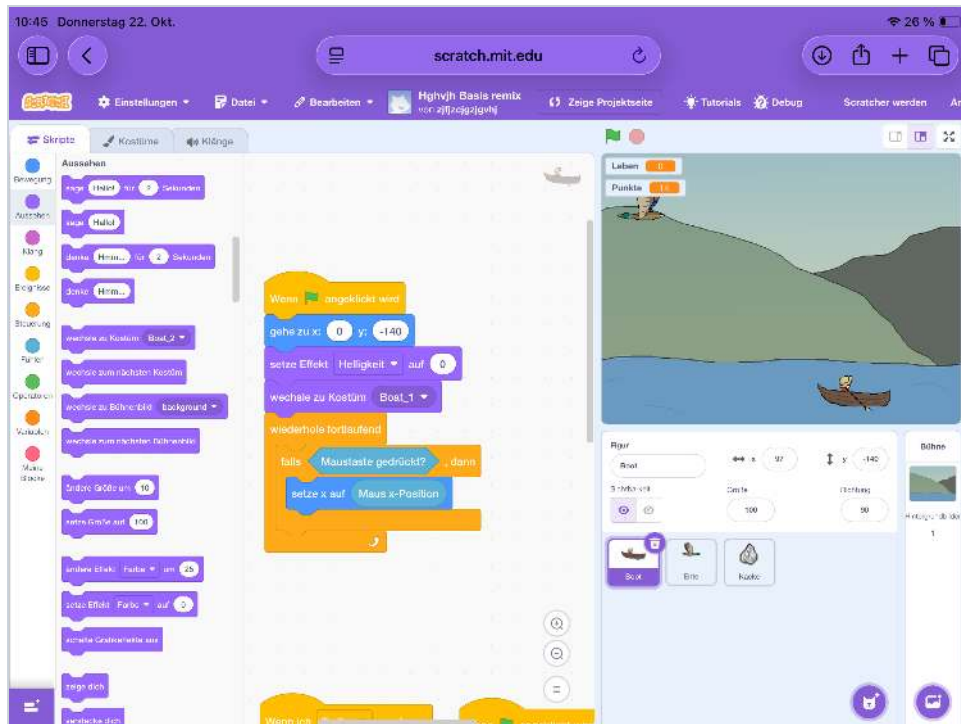
8.4 Sauberer Neustart (3 min)

Wenn du nach einem Treffer sofort das Spiel neu startest, ist das Boot manchmal noch halb durchsichtig oder hat das falsche Kostüm. Wir setzen am Start alles zurück.

Bleib beim **Boot**. Öffne das Steuer-Skript aus Block 2 und füge nach „gehe zu x: 0 y: -140“ zwei Blöcke ein:

```
Wenn  angeklickt wird
  gehe zu x: (0) y: (-140)
  setze Effekt [Durchsichtigkeit] auf (0)
  wechsle zu Kostüm [Boot1]
  wiederhole fortlaufend
    falls <Maustaste gedrückt?> dann
      gehe zu x: (Maus x-Position) y: (-140)
```

Beide Blöcke kommen aus **Aussehen** (lila).



Was ist passiert?

Alles, was du bei „Wenn  angeklickt wird“ **vor** der Schleife setzt, ist der **Startzustand**. Ein Spiel braucht einen klaren Startzustand: Position, Aussehen, Variablen. Sonst schleppst du Reste vom letzten Durchgang mit.

Merke: Alles, was sich im Spiel verändert, muss am Start einmal zurückgesetzt werden.

8.5 Speichern (1 min)

„**Jetzt speichern**“ tippen. Kontrolle: „Projekt gespeichert“ oben in der Leiste.

 **Leitung:** Nach diesem Block ist das Basis-Spiel komplett. Zwischenstand „Entenspiel Stand 08 – Feinschliff“ liegt im Studio für alle, die zurückliegen und in die Kreativzeit mit einem vollständigen Spiel starten wollen.

Herausforderung (wenn du schneller bist)

1. **Highscore:** Lege eine neue Variable **Highscore** an (Für alle Figuren). Im Game-Over-Skript des Boots **vor** dem „sage“-Block:

```
falls <(Punkte) > (Highscore)> dann
  setze [Highscore] auf (Punkte)
```

Der Highscore wird **nicht** bei der grünen Flagge auf 0 gesetzt, sonst ist er sinnlos. Häkchen bei Highscore in der Palette, damit er auf der Bühne steht.

2. **Startbildschirm:** Das Spiel soll erst losgehen, wenn du die Bühne antippst. Beim **Boot** ganz oben im Steuer-Skript vor der Schleife:

```
sage (Tippe auf die Bühne und weiche den Enten aus!) für (2) Sekunden
```

Und bei **Ente** und **Bühne** in jedes Skript mit einer Schleife direkt nach dem Hut-Block:

```
warte bis <Maustaste gedrückt?>
```

„warte bis“ ist in **Steuerung**, „Maustaste gedrückt?“ in **Fühlen**.

3. **Blinkende Ente:** Beim Treffer soll auch die Ente kurz die Farbe wechseln. Bei der Ente: „Wenn ich [Treffer] empfangen“ → „setze Effekt [Farbe] auf (50)“ → „warte (0.2) Sekunden“ → „setze Effekt [Farbe] auf (0)“. Eine Nachricht, mehrere Empfänger.

Checkliste Block 8

- ☐ Die Musik läuft in Dauerschleife.
- ☐ Alle 10 Punkte kommt ein Sound.
- ☐ Beim Game Over zeigt das Boot die Punktzahl.
- ☐ Nach Neustart ist das Boot wieder voll sichtbar und hat Kostüm Boot1.
- ☐ Ich habe gespeichert.

Weiter geht's: Jetzt bist du dran. In Block 9 baust du deine eigenen Ideen ein.

🎓 Leitung – Ablauf Block 8 (15 min)

Min	Was
0–1	Ziel zeigen: fertiges Spiel mit Musik, Pling, Game-Over-Text am Beamer
1–4	Musik-Loop auf der Bühne, Knatter-Fehler zeigen
4–8	mod erklären mit Tabelle, Bedingung gemeinsam zusammenbauen
8–11	verknüpfe im Game-Over-Skript
11–14	Reset-Blöcke, Konzept Startzustand
14–15	Speichern, Ausblick Kreativzeit

Vorbereiten: Zwischenstand „Entenspiel Stand 08 – Feinschliff“ im Studio. Klänge music_loop und point liegen bei der Bühne, hit beim Boot. Beispieltabelle für mod als Folie.

Typische Probleme: „spiele Klang“ statt „spiele Klang ganz“ → Knattern · „spiele Klang ganz“ beim Pling → Punkte-Schleife stockt eine Sekunde · verschachtelte Operatoren rutschen beim Ziehen auseinander · Klang point beim Boot gesucht statt auf der Bühne · Highscore versehentlich am Start auf 0 gesetzt.

Block 9: Kreativzeit (35 min)

Teilnehmer-Handout. Kästen mit 🎓 sind Hinweise für die Workshop-Leitung und werden vor dem Druck entfernt.

Was du in diesem Block lernst

- Eine eigene Idee in Blöcke übersetzen
- Mit dem, was du kennst, etwas Neues bauen
- Ausprobieren, testen, verbessern

Dein Spiel läuft. Jetzt gehört es dir. Die Leitung zeigt dir ein paar fertige **Ideenkarten**. Danach hast du **drei Minuten Ruhe**, um dir deine eigene Idee zu überlegen: Was fehlt deinem Spiel? Schreib sie auf die leere Ideenkarte, die du bekommst. Wer mag, stellt seine Idee danach kurz vor. Niemand muss.

Keine eigene Idee? Kein Problem, dann nimm eine von den fertigen Karten auf den nächsten Seiten.

9.1 So gehst du vor (2 min)

1. **Eine Idee wählen.** Lieber eine kleine Idee fertig als drei halbe. Fang mit ★ oder ★★ an.
2. **Klein anfangen.** Erst der einfachste Teil. Beispiel Power-up: erst fallen lassen, dann Treffer, dann Leben +1.
3. **Oft testen.** Nach jedem neuen Block die grüne Flagge drücken.
4. **Abschauen erlaubt.** Vieles, was du brauchst, hast du in Block 2 bis 8 schon gebaut. Die Karten sagen dir, wo.
5. **Speichern.** Alle 5 Minuten „Jetzt speichern“.

Die Sterne sagen, wie schwer eine Idee ist:

★	Geht schnell, wenig neue Blöcke
★★	Braucht mehrere Blöcke und etwas Nachdenken
★★★	Für Tüftler, geht über die Kreativzeit hinaus

Alle Ideen auf einen Blick:

Idee	Schwierigkeit	Was passiert
 Treffer hörbar machen	★	Geräusch beim Treffer, auch selbst aufgenommen
 Wolken am Himmel	★	Eine Wolke zieht langsam hinter der Ente vorbei
 Herz-Power-up	★★	Manchmal fällt ein Herz: Leben +1
 Highscore	★★	Bester Durchgang bleibt stehen
 Kacke wird schneller	★★	Kacke fällt mit der Zeit schneller
 Schwierigkeit wählen	★★	Spiel fragt am Anfang: leicht, mittel, schwer?
 Zweite Ente	★★★★	Nach 30 Sekunden kommt eine zweite Ente
 Pause-Knopf	★★★★	Knopf hält das Spiel an und startet es wieder
 Deine eigene Idee	–	Was fehlt deinem Spiel?

🎓 **Leitung:** Reihenfolge ist wichtig. **Erst** die fertigen Ideen am Beamer zeigen (drei Folien ★/★★/★★★, je eine Minute), damit alle wissen, was in der Zeit machbar ist. **Dann** drei Minuten Stille: jede/r überlegt die eigene Idee und schreibt sie auf die leere Ideenkarte (Name, Idee, erste Schritte). Nicht reden, nicht am iPad. **Danach** freiwillig vorstellen: „Wer mag, sagt in einem Satz, was er oder sie baut.“ Zwei, drei Wortmeldungen reichen, sie bringen die anderen auf Ideen. Niemanden aufrufen. Wer keine eigene Idee hat, nimmt eine fertige Karte. Das verhindert 10 Minuten Herumklicken.

9.2 Ideenkarten

Treffer hörbar machen ★

Bisher ist ein Treffer stumm. Gib ihm ein Geräusch.

Blöcke, die du brauchst: „spiele Klang ()“ aus **Klang** (pink).

So geht's: Boot wählen → Skript „**Wenn ich Treffer empfangen**“ → den Klang-Block direkt unter „**ändere Leben um -1**“ einhängen, über die Blink-Schleife. Wähle im Dropdown **hit** — der Klang liegt schon beim Boot.

Und dann dein eigener: Tab „Klänge“ → unten links auf den Lautsprecher-Knopf halten → „**Aufnehmen**“ (Mikrofon). Safari fragt einmal nach Erlaubnis. Nimm dein eigenes „Autsch!“ auf, schneide es zu, speichere — und wähle es im Klang-Block statt „hit“.

Tipp: „spiele Klang ()“ läuft weiter, während das Boot blinkt. „spiele Klang () ganz“ wartet erst das Geräusch ab — dann blinkt das Boot später. Probier beides.

Wo nachschauen: Block 6 (Treffer-Skript des Boots)

Wolken am Himmel ★

Der Himmel ist leer. Eine Wolke, die langsam von rechts nach links zieht, macht das Bild lebendig.

Blöcke, die du brauchst: „gehe ()er Schritt“, „pralle vom Rand ab“, „setze Drehtyp auf links-rechts“ aus **Bewegung**, „wiederhole fortlaufend“ aus **Steuerung**.

Tipp: Neue Figur über den Katzen-Knopf unten rechts → „Figur wählen“ → „Cloud“ suchen. Skript wie die Ente aus Block 4, aber „gehe (1)er Schritt“ und ganz oben „gehe zu Ebene (hinten)“ aus **Aussehen**, damit die Wolke hinter der Ente bleibt. Die Wolke darf **nicht** das Boot treffen können, also hoch platzieren: y = 160.

Wo nachschauen: Block 4 (Ente fliegt)

Herz-Power-up ★★

Manchmal fällt statt Kacke ein Herz. Fängst du es, bekommst du ein Leben zurück.

Blöcke, die du brauchst: „erzeuge Klon von ()“, „Wenn ich als Klon entstehe“, „lösche diesen Klon“ aus **Steuerung**, „wird (Boot) berührt?“ aus **Fühlen**, „ändere (Leben) um (1)“ aus **Variablen**, „Zufallszahl von () bis ()“ aus **Operatoren**.

Tipp: Neue Figur „Herz" aus der Bibliothek („Heart"). Skript fast wie die Kacke aus Block 5 und 6, aber bei „berührt Boot" → „ändere Leben um 1" statt Nachricht „Treffer". Damit das Herz seltener kommt, bei der Ente im Kack-Skript:

```
falls <(Zufallszahl von (1) bis (5)) = (1)> dann
  erzeuge Klon von [Herz]
sonst
  erzeuge Klon von [Kacke]
```

Nicht vergessen: „Wenn  angeklickt wird → verstecke dich" auch beim Herz.

Wo nachschauen: Block 5 (Klone), Block 6 (Treffer)

Highscore ★★

Der beste Durchgang bleibt stehen, auch nach Neustart.

Blöcke, die du brauchst: neue Variable „Highscore", „falls < > dann", „() > ()".

Tipp: Steht als Herausforderung in Block 8. Wichtig: Highscore **nicht** bei der grünen Flagge auf 0 setzen.

Wo nachschauen: Block 8, Herausforderung 1

Kacke wird schneller ★★

Nicht nur mehr Kacke, sondern auch schnellere Kacke.

Blöcke, die du brauchst: Variable „Tempo" (aus Block 7, Herausforderung A), „() - ()" aus **Operatoren**.

Tipp: Im Klon-Skript der Kacke steht „ändere y um (-5)". Ersetze die -5 durch „(0) - (Tempo)". Wenn Tempo auf der Bühne pro Sekunde wächst, fällt die Kacke immer schneller. Falls du Tempo noch nicht hast: Block 7, Herausforderung A.

Wo nachschauen: Block 5 (Klon-Skript), Block 7 (Tempo)

Schwierigkeit wählen ★★

Am Anfang fragt das Spiel: leicht, mittel oder schwer?

Blöcke, die du brauchst: „frage () und warte" und „Antwort" aus **Fühlen**, „falls < > dann", „() = ()", „setze (Pause) auf ()".

Tipp: Auf der Bühne ganz oben im Punkte-Skript vor der Schleife:

```
frage (Leicht, mittel oder schwer?) und warte
falls <(Antwort) = (schwer)> dann
  setze [Pause] auf (1)
```

Bei „leicht" Pause auf 3. Die Frage öffnet auf dem iPad die Tastatur, das funktioniert.

Wo nachschauen: Block 7 (Variable Pause)

Zweite Ente ★★★★★

Nach 30 Sekunden kommt eine zweite Ente dazu.

Blöcke, die du brauchst: Figur duplizieren, „warte (30) Sekunden", „verstecke dich" / „zeige dich".

Tipp: Steht als Herausforderung B in Block 7. Wichtig: Auch die Kacke duplizieren und in Kacke2 „gehe zu (Ente2)" einstellen.

Wo nachschauen: Block 7, Herausforderung B

Pause-Knopf ★★★★★

Ein Knopf, der das Spiel anhält und wieder startet.

Blöcke, die du brauchst: neue Figur „Pause", neue Variable „Pausiert", „warte bis < >", „Wenn diese Figur angeklickt wird".

Tipp: Das ist knifflig, weil Scratch kein „alles anhalten und weitermachen“ hat. Weg: Variable „Pausiert“ (0 oder 1). Der Knopf schaltet sie um. In **jeder** Schleife (Boot, Ente, Kacke, Bühne) als erste Zeile „warte bis <(Pausiert) = (0)>“. Sprich mit der Leitung, bevor du anfängst.

Wo nachschauen: Block 6 (Variablen), Block 8 (warte bis)

💡 Deine eigene Idee

Was fehlt deinem Spiel? Schreib es hier auf, dann zerlege es in kleine Schritte:

Idee: _____

Schritt 1: _____

Schritt 2: _____

Schritt 3: _____

Welche Blöcke kenne ich schon, die dabei helfen? _____

🎓 **Leitung:** Nach 15 Minuten Zeit-Check am Beamer: „Wer hat schon etwas Neues im Spiel? Zeig kurz.“ Dabei Speichern-Erinnerung. Wer hängt, wird auf eine ★-Karte umgelenkt, nicht auf Hilfe warten lassen. Schnelle bekommen eine zweite Karte oder helfen einem Nachbarn (klar sagen: erklären, nicht selbst tippen). Fünf Minuten vor Schluss ansagen, damit das Spiel in einem laufenden Zustand gespeichert wird. Kaputte Experimente: Blöcke rausziehen, nicht das Projekt neu laden. Scratch-Bibliothek: Figuren über den Katzen-Knopf unten rechts in der Figurenliste, Klänge über den Lautsprecher-Knopf im Tab „Klänge“. Beides braucht Internet. Malwerkzeug auf dem iPad: Finger funktioniert, Formen und Füllen sind einfacher als Freihand. Der Bitmap/Vektor-Umschalter unten links im Kostüm-Editor verwirrt, am besten bei Vektor lassen.

Checkliste Block 9

- ☐ Ich habe mir eine Idee ausgesucht.
- ☐ Mindestens ein Teil davon funktioniert im Spiel.
- ☐ Das Spiel läuft noch ohne Fehler durch (grüne Flagge bis Game Over).
- ☐ Ich habe gespeichert.

Weiter geht's: In Block 10 zeigst du dein Spiel den anderen.

🎓 Leitung – Ablauf Block 9 (35 min)

Min	Was
0–3	Fertige Ideen am Beamer zeigen (★, ★★, ★★★), leere Ideenkarten austeilen
3–6	Stille Denkzeit: eigene Idee auf die Ideenkarte schreiben
6–9	Freiwillige stellen ihre Idee in einem Satz vor
9–17	Bauen, Betreuer gehen herum
17–19	Zeit-Check, 2–3 Zwischenstände zeigen, Speichern
19–30	Bauen
30–32	Ansage „5 Minuten“, Spiel in lauffähigen Zustand bringen
32–35	Speichern, aufräumen, kurz überlegen: Was zeige ich gleich?

Rollen: Fach-Betreuer nimmt die ★★★-Ideen und Fehlersuche. Orga-Betreuer hält Zeit, erinnert an Speichern, hilft bei Bibliothek und Malwerkzeug.

Vorbereiten: Leere Ideenkarten gedruckt und zugeschnitten (scripts/build.sh ideas → build/ideenkarten.pdf, eine je Person plus Reserve), Stifte, Zwischenstand „Entenspiel Stand 08 – Feinschliff“ im Studio für alle, die hinterher sind. Eine Demo mit Herz-Power-up und Level-Farbe im Lehreraccount, um am Beamer zeigen zu können, wohin es gehen kann.

Typische Probleme: Zu große Idee gewählt und nach 20 Minuten nichts läuft · Spiel durch Experimente kaputt und nicht gespeichert · neue Figur aus Bibliothek riesig (Größe im Figurenfeld auf 30–50 setzen) · Klon-Figur nicht am Start versteckt · Variable „nur für diese Figur“ angelegt.

Block 10: Präsentation & Abschluss (15 min)

Teilnehmer-Handout. Kästen mit  sind Hinweise für die Workshop-Leitung und werden vor dem Druck entfernt. Platzhalter für Bilder:  [Bild: ...]

Was du in diesem Block lernst

- Dein Spiel als Datei sichern und mit nach Hause nehmen
- Dein Spiel in einer Minute vorstellen
- Wie du zu Hause weitermachst

10.1 Spiel speichern und als Datei sichern (4 min)

Dein Scratch-Konto aus dem Workshop ist ein **Schüler-Konto**. Es wird nach dem Kurs geschlossen, und dein Projekt darin verschwindet. Dein Spiel bleibt dir nur als **Datei**. Deshalb:

1. „**Jetzt speichern**“ tippen.
2. Im Editor oben links auf „**Datei**“ tippen.
3. „**Auf deinen Computer herunterladen**“ wählen.
4. Safari fragt: „**Laden**“ tippen.
5. Die Datei heißt Entenspiel <Benutzername>.sb3 und liegt im Safari-Download-Ordner. Du findest sie in der App „**Dateien**“ unter „Downloads“.
6. Die Datei an den **Laptop der Leitung** schicken: In „Dateien“ die Datei lange drücken → „**Teilen**“ → **AirDrop** → den Laptop auswählen. Die Leitung sagt dir, wie er heißt.

Diese .sb3-Datei kannst du auf jedem Computer mit Scratch öffnen: scratch.mit.edu → Entwickeln → Datei → „Von deinem Computer hochladen“. Du bekommst sie von uns nach dem Workshop.

 **Leitung:** Die Projekte werden **nicht** geteilt (Entscheidung: kein öffentliches Teilen). Deshalb ist die Datei der einzige Weg nach Hause. Orga-Betreuer nimmt jede .sb3 per AirDrop auf dem Laptop entgegen und hakt auf der Teilnehmer-Karte ab, wessen Datei da ist. Nach dem Workshop die Datei an die Mailadresse aus der Anmeldung schicken. Falls AirDrop auf den iPads gesperrt ist: Download bleibt in „Dateien“, und die Leitung lädt die Projekte selbst aus der Klassenübersicht herunter (Projekt öffnen → Datei → „Auf deinen Computer herunterladen“), bevor die Klasse beendet wird.

10.2 Dein Spiel vorstellen (8 min)

Wer will, zeigt sein Spiel. Niemand muss. Du spielst es **von deinem iPad aus** am Beamer vor, kurz, und beantwortest drei Fragen:

1. **Was habe ich Neues eingebaut?** Zeig es im Spiel oder im Skript.
2. **Was war schwer?** Und wie hast du es gelöst?
3. **Was würde ich als Nächstes bauen?**

So kommt dein iPad auf den Beamer (Apple TV):

1. In Scratch die **grüne Flagge** bereit, am besten die Bühne im **Vollbild** (Knopf rechts über der Bühne).
2. Vom **rechten oberen Rand** nach unten wischen: das Kontrollzentrum öffnet sich.
3. Auf „**Bildschirmsynchronisation**“ tippen (das Symbol mit den zwei Rechtecken).
4. Das **Apple TV** aus der Liste wählen. Die Leitung sagt dir den Namen.
5. Falls ein **Code** auf der Leinwand erscheint: auf dem iPad eintippen.
6. Spielen. Danach wieder Kontrollzentrum → „Bildschirmsynchronisation“ → „**Synchronisation stoppen**“, damit der Nächste drankommt.

Das ist keine Prüfung. Alle haben heute das gleiche Spiel gebaut und wissen, wie viel Arbeit darin steckt. Klatschen nach jedem Spiel ist Pflicht.

Während du zuhörst: Merk dir eine Idee aus einem anderen Spiel, die du selbst noch einbauen willst.

🎓 Leitung: Freiwillig, niemanden aufrufen. Zwei, drei Vorstellungen sind genug; wer danach noch will, darf. Das iPad geht per **AirPlay auf das Apple TV**, der Beamer hängt dafür am Apple TV (Eingang umschalten, nicht mehr am Laptop). Vorteil: Finger-Steuerung bleibt, das Kind spielt selbst, nichts muss gespeichert oder geöffnet sein. Vor dem Block mit einem Teilnehmer-iPad testen: gleiches WLAN wie das Apple TV, AirPlay-Code sichtbar, Ton kommt mit. Den Apple-TV-Namen an die Tafel schreiben. Zwischen zwei Vorstellungen stoppt das erste iPad die Synchronisation, sonst sieht der Nächste das Gerät nicht als frei. Wer nicht vor der Gruppe reden will, spielt nur und die Leitung stellt die Fragen. Orga-Betreuer hält sichtbar die Zeit, eine Minute pro Spiel, freundlich, aber konsequent. **Rückfall**, wenn AirPlay nicht geht: Laptop an den Beamer, Projekt aus der **Klassenübersicht** des Lehreraccounts öffnen (Meine Klassen → Klasse → Schüler → Name → Projekte); dann muss das Projekt vorher „Jetzt gespeichert“ sein.

10.3 Zu Hause weitermachen (1 min)

Du kannst jederzeit weitermachen:

- **scratch.mit.edu** im Browser, am Computer oder Tablet. Kostenlos.
- Lege dir mit einem Erwachsenen ein **eigenes Scratch-Konto** an. Dein Workshop-Konto ist nur für den Kurs.
- Öffne deine .sb3-Datei über **„Datei → Von deinem Computer hochladen“**. Mit eigenem Konto kannst du sie dort speichern und weiterbauen.
- Das Studio **„Entenpaddeln-Basis“** über den Link auf deinem Zettel enthält das Startprojekt und alle Zwischenstände. Falls deine Datei verloren geht, remixt du den letzten Stand und baust deine Erweiterung noch einmal ein.
- Ideen für später: die ★★-Karten aus Block 9.

 [Bild: Zettel zum Mitnehmen: Studio-Link als QR-Code, kurze Anleitung Konto + Datei hochladen]

10.4 Feedback (1 min)

Drei Fragen, ehrlich beantwortet, hilft uns für den nächsten Workshop:

	😊	😐	😞
Ich habe verstanden, was ich gebaut habe			
Das Tempo war richtig			
Ich will weitermachen			

Was hat dir am meisten Spaß gemacht? _____

Was war unnötig oder zu schwer? _____

🎓 Leitung: Feedback auf Papier oder per Daumen hoch/mitte/runter im Raum, je nach Zeit. Die freien Antworten sind für die Überarbeitung der Blöcke am wertvollsten. Antworten nicht kommentieren, nur danken.

Checkliste Block 10

- ☐ Ich habe „Jetzt speichern“ getippt.
- ☐ Ich habe die .sb3-Datei heruntergeladen und an den Laptop geschickt.
- ☐ Ich habe mein Spiel vorgestellt, oder ich habe zugeschaut und mir eine Idee gemerkt.
- ☐ Ich weiß, wie ich zu Hause weitermache.

Geschafft. Du hast heute ein Spiel von null auf fertig programmiert: Steuerung, Animation, Klone, Variablen, Nachrichten, Schwierigkeitskurve. Das sind dieselben Bausteine, aus denen große Spiele bestehen.

Leitung – Ablauf Block 10 (15 min)

Min	Was
0–4	Gemeinsam speichern, .sb3 herunterladen, per AirDrop an den Laptop; Orga-Betreuer hakt ab
4–12	Freiwillige Präsentationen vom iPad über Apple TV, je etwa 1 min
12–13	Zettel zum Mitnehmen, Weitermachen zu Hause
13–15	Feedback, Verabschiedung

Rollen: Fach-Betreuer moderiert die Präsentationen und stellt die drei Fragen. Orga-Betreuer hält Zeit, hilft beim AirPlay-Verbinden und -Trennen, nimmt die AirDrop-Dateien am Laptop an, hilft beim Herunterladen, verteilt Zettel und Feedbackbögen.

Vorbereiten: Apple TV am Beamer, Beamer-Eingang auf Apple TV umschaltbar, Apple TV im selben WLAN wie die iPads, AirPlay-Code aktiv (sonst spiegelt irgendwann ein falsches iPad), Apple-TV-Name an der Tafel, Verbindung mit einem Teilnehmer-iPad getestet, Ton geprüft. Zettel zum Mitnehmen mit Studio-Link als QR-Code und Kurzanleitung „eigenes Konto anlegen, Datei hochladen“. Feedbackkarten gedruckt und zugeschnitten (scripts/build.sh feedback → build/feedbackkarten.pdf, eine je Person). Laptop als Rückfall mit Lehreraccount in der Klassenübersicht eingeloggt, AirDrop am Laptop auf „Für alle“ gestellt und Gerätenamen bekannt. Entscheidung steht: **kein öffentliches Teilen**, keine Teilnehmerprojekte im Studio.

Typische Probleme: Download landet nicht sichtbar, „Dateien → Downloads“ zeigen · AirDrop findet den Laptop nicht (WLAN und Bluetooth am Laptop an, Laptop entsperrt) · Apple TV erscheint nicht in der Liste (iPad in anderem WLAN, oder voriges iPad hat die Synchronisation nicht gestoppt) · Beamer zeigt noch den Laptop (Eingang umschalten) · Bild da, kein Ton (Lautstärke am iPad, Beamer-Lautsprecher) · Safari-Leisten stören (Bühne auf Vollbild) · Zeit läuft bei den Präsentationen weg, dann Fragen kürzen · Teilnehmer wollen Schüler-Konto behalten: erklären, dass es temporär ist · jemand tippt doch auf „Teilen“: Projekt auf der Projektseite wieder auf „Nicht teilen“ setzen.