Einführung für Lehrkräfte

> **Fassung ohne Lösungen:** Dieses Handbuch enthält keine Musterlösungen. Die Lösungen und die genannten Programmdateien gibt es für Lehrkräfte auf Anfrage per E-Mail an info@picoboy.de.

Dieses Heft erklärt, was Sie vor der ersten Stunde wissen und vorbereiten sollten: das Gerät, die Programmierumgebung PBC+ Studio, die Wahl zwischen Emulator und Gerät, die Einrichtung im Computerraum und das Aufspielen der Firmware.

Auf einen Blick

Gerät	PicoBoy Color Plus (kurz PBC+): Farbbildschirm 240 × 280 Pixel, Joystick (fünf Richtungen inkl. Drücken = MITTE), Tasten A und B, drei Status-LEDs (rot, gelb, grün), eine RGB-LED, Lautsprecher, Beschleunigungssensor, USB-C
Programmierumgebung	PBC+ Studio im Browser: **pbcstudio.de**, Chrome oder Edge, Build 109 oder neuer
Sprache im Unterricht	nur Blöcke. Die Blöcke werden intern in MicroPython übersetzt; das müssen die Schülerinnen und Schüler nicht sehen.
Ohne Gerät?	Ja. Der eingebaute Emulator führt alle Programme aus – bis auf den Beschleunigungssensor (siehe unten).
Anmeldung	keine. Kein Konto, keine Klassenverwaltung.
Material	23 Arbeitsblätter (AB 01–AB 23), je ein Lehrermaterial mit Stundenverlauf und typischen Fehlern, das Schülerbuch, dazu L 01–L 07

Das Gerät

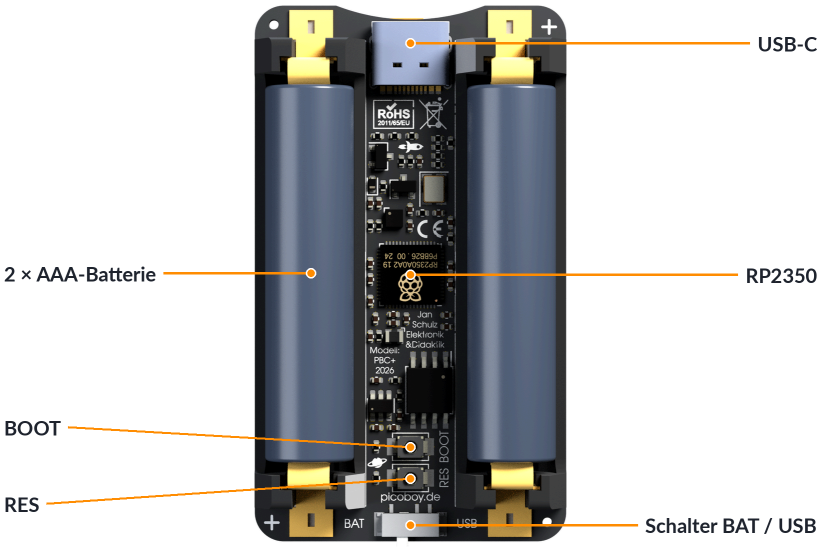
Der PicoBoy Color Plus ist ein kleiner Computer im Format einer Handkonsole. Im Inneren arbeitet ein Mikrocontroller RP2350 mit MicroPython als Firmware. Die Schülerinnen und Schüler programmieren ihn mit Blöcken; PBC+ Studio schickt das Programm über das USB-Kabel aufs Gerät.



- **Eingabe:** Joystick hoch/runter/links/rechts, Joystick drücken (= MITTE), Taste A, Taste B, Beschleunigungssensor (Kippen)
- **Ausgabe:** Bildschirm, drei Status-LEDs, RGB-LED, Lautsprecher

Dieses Handbuch wurde mit künstlicher Intelligenz (KI) generiert. Fehler sind möglich.

- **Rückseite:** zwei AAA-Batterien, der Schalter BAT/USB (BAT = Strom aus den Batterien, USB = vom Kabel; auf USB ohne Kabel ist das Gerät aus) und direkt darüber zwei kleine Knöpfe **RES** (Neustart) und **BOOT** (zum Aufspielen der Firmware, siehe unten)

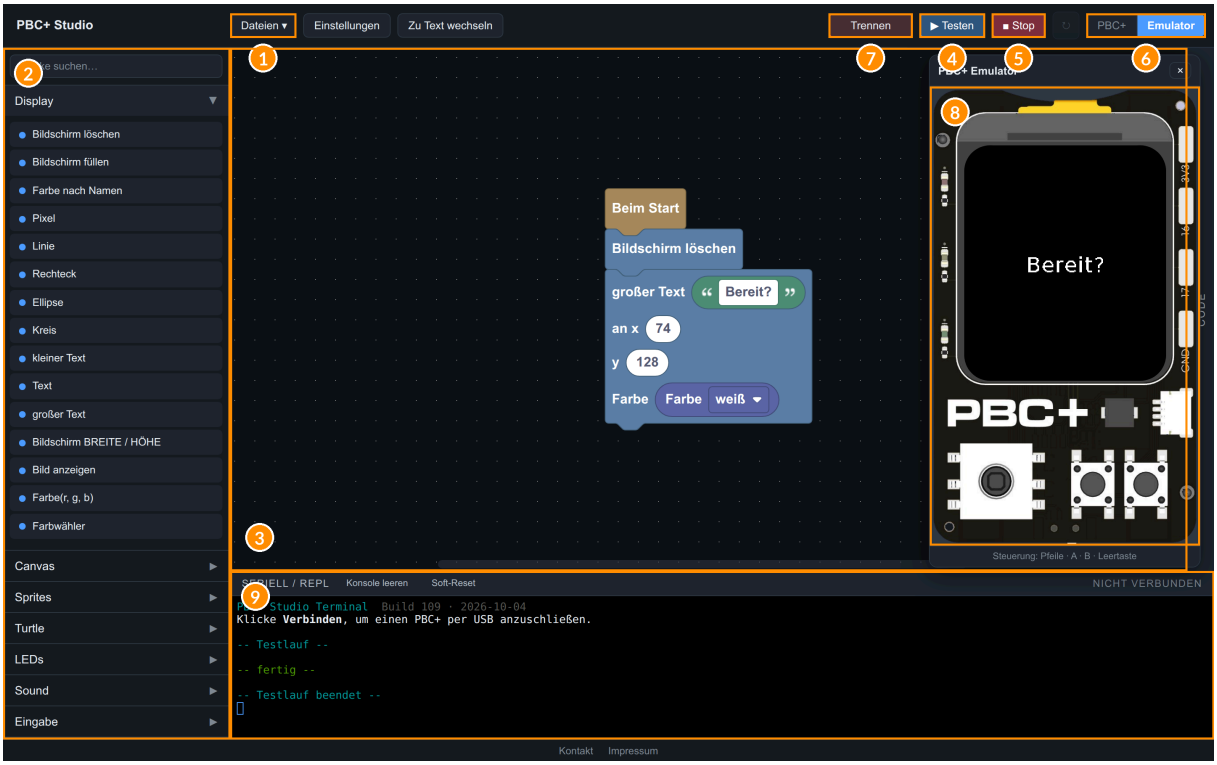


Das Koordinatensystem des Bildschirms beginnt **oben links** bei (0, 0); x wächst nach rechts, y nach **unten**. Die Turtle-Grafik verwendet dagegen ein eigenes Koordinatensystem mit (0, 0) in der Mitte und y nach oben. Kopiervorlagen für beide finden Sie in L 07.

> **Umlaute:** Die Schriften auf dem Bildschirm kennen nur Zeichen ohne Umlaute. „Grün“ erscheint als „Gr?n“ – im Emulator genauso wie auf dem Gerät. Lassen Sie in Bildschirmtexten „ae, oe, ue, ss“ schreiben. Auch Namen von Variablen, Listen, Funktionen und Sprites dürfen keine Umlaute enthalten; PBC+ Studio lehnt sie ab.

PBC+ Studio

PBC+ Studio ist eine einzelne Webseite. Sie läuft vollständig im Browser; eine Installation ist nicht nötig. Öffnen Sie **pbcstudio.de** in **Chrome** oder **Edge**. Nur diese Browser können über die Web-Serial-Schnittstelle mit dem Gerät sprechen. Welche Version läuft, steht beim Start im Terminal unten („Build 109 · 2026-10-04“). Das Material bezieht sich auf **Build 109**; neuere Builds sollten passen, kleine Abweichungen in Beschriftungen sind möglich.



Nr.	Bereich	Hinweis für die Lehrkraft
1	**Dateien**	Speichern/Öffnen, Bilder, Firmware, Startmenü – siehe unten
2	**Blockleiste**	Kategorien *Display*, *Canvas*, *Sprites*, *Turtle*, *LEDs*, *Sound*, *Eingabe*, *Sensor*, *Kontrollstrukturen*, *Variablen*, *Listen*, *Funktionen*, *Mathe*, *Text*, *Zeit*, *Konsole*. Oben ein Suchfeld. *Externe Sensoren* erscheint nur, wenn in den Einstellungen ein Zusatzsensor aktiviert ist; das Material braucht sie nicht.
3	**Arbeitsfläche**	Nur Blöcke unter „Beim Start“ werden ausgeführt. **Strg+Z** macht rückgängig.
4 / 5	**▶ Testen / ■ Stop**	startet bzw. beendet das Programm im gewählten Ziel
6	**Ziel**	**Emulator** oder **PBC+** (echtes Gerät)
7	**Verbinden / Trennen**	stellt die USB-Verbindung her
8	**Emulator**	der nachgebildete PicoBoy
9	**Terminal**	Meldungen und Fehlermeldungen

Der Knopf **Zu Text wechseln** zeigt den erzeugten Python-Code. Das Material braucht ihn nicht. Für Interessierte ist er ein schöner Ausblick, aber Vorsicht: Wer im Text-Modus Code ändert und dann zu den Blöcken zurückwechselt, verliert diese Änderungen (PBC+ Studio fragt vorher nach).

Emulator oder Gerät?

Das **Ziel** oben rechts entscheidet, wo ein Programm läuft. Beide Wege funktionieren mit denselben Blöcken.

	Emulator	Gerät (PBC+)
Voraussetzung	nur ein Browser mit Internetzugang	Gerät mit MicroPython-Firmware, USB-Datenkabel, Chrome/Edge
Bedienung	Pfeiltasten = Joystick, Leertaste = MITTE, Tasten **A** und **B**; die Knöpfe lassen sich auch mit der Maus anklicken	echte Tasten
Ton	über die Lautsprecher/Kopfhörer des Rechners	Lautsprecher im Gerät
Bilder	werden im Browser gespeichert	werden auf dem Gerät gespeichert
Kippen	nicht möglich (siehe Grenzen)	ja
Programm behalten	als Datei auf dem Rechner, zusätzlich mit **Auf den PBC+...** im Browser (ohne Startmenü)	zusätzlich auf dem Gerät mit Startmenü

Empfehlung: Entwickeln im Emulator, Ausprobieren auf dem Gerät. Der Emulator ist schnell startklar und braucht keine Kabel. Das Gerät motiviert stark – ein Programm in der Hand zu halten, ist etwas anderes als ein Bild auf dem Monitor. Wenn Sie nur wenige Geräte haben, arbeiten die Schülerinnen und Schüler im Emulator und testen fertige Stände an einer „Gerätestation“.

Grenzen des Emulators

- **Kein Beschleunigungssensor.** Der Emulator meldet immer ein flach auf dem Tisch liegendes Gerät: x = 0, y = 0, z = 1 g (9,81 m/s²). Programme mit „Beschleunigung“ laufen ohne Fehler, reagieren aber nicht. AB 21 („Kipp mich!“) braucht deshalb Geräte.
- **Bemalter Canvas ist langsamer.** Wird ein Canvas mit vielen Zeichenblöcken bemalt, dauert das im Emulator spürbar länger als auf dem Gerät. Ein Hintergrundbild („Bild auf Canvas“) ist dagegen schnell.
- **Bildschirmzeit nachgebildet.** Der Emulator wartet beim Anzeigen etwa so lange wie das echte Display (rund 40 ms für ein ganzes Bild). Flackern und Ruckeln sehen deshalb ähnlich aus wie auf dem Gerät – das ist gewollt.
- **Endfassung auf dem Gerät prüfen.** Geschwindigkeit und Tastengefühl unterscheiden sich leicht. Spiele, die vorgeführt werden sollen, vorher einmal auf dem Gerät testen.

Einrichtung im Computerraum

Netzwerk. PBC+ Studio lädt beim Start Programmbibliotheken nach: Blockly und CodeMirror von `cdnjs.cloudflare.com`, das Terminal (xterm) von `cdn.jsdelivr.net`. Beim ersten Start des Emulators kommt MicroPython als WebAssembly (rund 700 KB) ebenfalls von `cdn.jsdelivr.net`. Ist ein Webfilter aktiv, müssen **pbcstudio.de**, **cdnjs.cloudflare.com** und **cdn.jsdelivr.net** freigegeben sein. Testen Sie das einmal auf einem Schülerrechner mit einem Schülerkonto.

Was der Browser speichert. PBC+ Studio merkt sich das zuletzt bearbeitete Programm, das gewählte Ziel und die Einstellungen im Speicher des Browsers (localStorage). Bilder für den Emulator liegen ebenfalls im Browser (IndexedDB). Das hat Folgen:

- An einem geteilten Rechner findet die nächste Klasse das Programm der vorherigen vor. **Dateien → Neu** setzt die Arbeitsfläche zurück.
- Werden Browserdaten beim Abmelden gelöscht (z. B. bei servergespeicherten oder zurückgesetzten Profilen), sind Programm **und hochgeladene Bilder** weg.
- Gespeichert ist es nur in diesem Browser auf diesem Rechner.

Sichern als Datei. Gewöhnen Sie die Klasse von Anfang an daran, am Ende jeder Stunde **Dateien → Auf den Computer…** zu wählen. Der Browser legt dann `programm.pbcp` im Download-Ordner ab (bei Wiederholung `programm (1).pbcp` usw.). Sinnvoll sind ein eigener Ordner je Klasse im Netzlaufwerk und sprechende Namen wie `ab06_ball_mia.pbcp`. Stellen Sie im Browser ggf. ein, dass vor dem Herunterladen nach dem Speicherort gefragt wird. Geöffnet wird mit **Dateien → Vom Computer…**.

> Eine `.pbcp`-Datei enthält die Blöcke und die Namen der Sprites, aber **keine Bilder**. Wer ein Projekt mit Bildern an einem anderen Rechner fortsetzt, muss die PNG-Dateien dort erneut hochladen. Legen Sie die Bilder für die Arbeitsblätter deshalb zentral ab (z. B. Tauschordner).

Bilder hochladen. Dateien → Bilder verwalten… zeigt die Bilder des **gerade gewählten Ziels**: bei *Emulator* die Bilder im Browser, bei *PBC+* die Bilder auf dem verbundenen Gerät. Neue Bilder (PNG) lassen sich dort hochladen, mehrere auf einmal mit **ZIP hochladen…**; ein Bildeditor hilft beim Zuschneiden auf die Bildschirmgröße. Mit **Dateien → Bild zeichnen…** malen die Schülerinnen und Schüler eigene Bilder. Für Sprites gilt: Das Bild muss hochgeladen sein, bevor man in der Kategorie **Sprites** „+ neues Sprite…“ anlegt. Die Bilder zu den Arbeitsblättern liegen im Ordner `dokumentation/material`.

Geräte vorbereiten

1. Firmware aufspielen (einmalig je Gerät)

Damit PBC+ Studio mit dem Gerät arbeiten kann, muss **MicroPython für den PBC+** darauf laufen. Auf das Gerät passt immer nur **eine** Firmware. Wurde zwischendurch ein Spiel aufgespielt, muss MicroPython danach wieder drauf.

1. Firmware-Datei bereitlegen: `micropython-PBC_PLUS.uf2` (im Projekt unter `pbc-plus/dist/`) oder über **Dateien → MicroPython installieren…** in PBC+ Studio herunterladen (dort heißt die Datei `micropythonPBCP.uf2`).
2. Gerät in den Flash-Modus bringen: **BOOT** gedrückt halten und dabei das USB-Kabel einstecken. Alternativ bei eingestecktem Kabel **BOOT** halten, **RES** kurz drücken, dann BOOT loslassen.
3. Am Rechner erscheint ein neues USB-Laufwerk (Name z. B. `RP2350` oder `RPI-RP2`).
4. Die `.uf2`-Datei auf dieses Laufwerk ziehen. Sobald das Kopieren fertig ist, verschwindet das Laufwerk und das Gerät startet neu.
5. Zeigt das Display danach noch ein altes Bild: Kabel kurz abziehen und wieder einstecken.

Der Menüpunkt **Dateien → MicroPython installieren…** spielt nichts selbstständig auf. Er öffnet eine Schritt-für-Schritt-Anleitung (wie oben) mit einem Download-Link zur Firmware. PBC+ Studio warnt dort, dass beim Aufspielen **alle Programme, Bilder und Dateien auf dem Gerät gelöscht** werden können. Wer etwas behalten will, erstellt vorher mit **Dateien → Flash-Backup erstellen (.uf2)…** eine Sicherung.

> Verwenden Sie eine Firmware vom Stand **Oktober 2026 oder neuer**. Ältere Fassungen kennen die Kollisionsblöcke („berührt Sprite“, „berührt Farbe“) noch nicht, die ab AB 13 gebraucht werden.

2. Verbinden und testen

Ziel **PBC+** wählen, **Verbinden** klicken, im Auswahlfenster des Browsers das Gerät auswählen. Dann ein Programm mit ▶ **Testen** starten. Klappt die Verbindung nicht:

- **Kabel prüfen.** Viele USB-C-Kabel sind reine Ladekabel ohne Datenleitungen. Markieren Sie geprüfte Datenkabel.
- **Browser prüfen.** Firefox und Safari können sich nicht verbinden.
- **Nur ein Programm** darf die Schnittstelle benutzen. Ist z. B. Thonny oder ein zweiter Browser-Tab verbunden, schlägt die Verbindung fehl.
- Unter Linux brauchen Benutzer ggf. die Berechtigung für serielle Schnittstellen (meist Gruppe `dialout`).

3. Startmenü installieren (einmalig je Gerät)

Dateien → Startmenü installieren… (nur bei verbundenem Gerät) legt auf dem Gerät eine kleine Datei `main.py` an, die beim Einschalten ein Menü startet. Gibt es schon eine `main.py`, wird sie vorher in `main(2).py` umbenannt – es geht nichts verloren.

Das Menü listet alle Python-Programme, die mit **Dateien → Auf den PBC+…** auf das Gerät gespeichert wurden (ohne Endung `.py`); Block-Projekte (`.pbcp`) erscheinen dort nicht. Bedienung: Joystick hoch/runter wählt, **A, B oder MITTE** startet das Programm. Die Beschriftungen im Menü sind englisch („Press button to run“). Endet ein Programm, startet das Gerät neu und zeigt wieder das Menü. Programme mit „unendlich wiederholen“ enden nie – dann **RES** auf der Rückseite drücken.

> **Auf den PBC+…** bietet im Blockmodus zwei Arten an: **Python-Programm** (`.py`, Standard) startet auf dem Gerät, lässt sich aber **nicht** wieder als Blöcke öffnen. **Block-Projekt** (`.pbcp`) dient zum Mitnehmen und Weiterbauen, startet aber **nicht** auf dem Gerät und steht nicht im Startmenü. **Vom PBC+…** zeigt beide Arten und öffnet `.pbcp` als Blöcke, `.py` im Text-Modus. Beide Menüpunkte arbeiten auch mit dem Ziel Emulator.

4. Bilder auf die Geräte bringen

Programme mit Sprites oder Hintergrundbildern brauchen die Bilder **auf dem Gerät**. Zwei Wege:

- Ziel **PBC+**, Gerät verbinden, **Dateien → Bilder verwalten…** und die PNG-Dateien hochladen, mehrere auf einmal mit **ZIP hochladen…**.
- Bilder liegen schon im Emulator: Gerät verbinden, **Dateien → Emulator → PBC+…** und die Dateien auswählen.

Planen Sie dafür vor der Stunde Zeit ein, z. B. für AB 15–17 und AB 22. Aufräumen lässt sich mit **Dateien → Dateien verwalten…**. Dort entpackt **Datei(en) oder ZIP hochladen…** eine ZIP flach aufs Gerät; `boot.py`, `main.py`, `pbc.py` und doppelte Namen werden ausgelassen.

Organisation

- **Nummerieren** Sie Geräte und Kabel und geben Sie sie immer an dieselben Paare aus. Dann finden die Schülerinnen und Schüler ihre gespeicherten Programme im Startmenü wieder.
- **Ausgabe und Rücknahme** mit einer Liste (Gerät, Kabel, Name). Am Ende der Stunde Kabel aufrollen, Geräte zählen.
- **Ladezustand** der Geräte vor Unterrichtsbeginn prüfen. Am USB-Kabel angeschlossen läuft das Gerät auf jeden Fall.
- **Ton:** Im Computerraum Kopfhörer bereitlegen oder die Lautstärke absprechen. AB 10 (Musik) ist sonst sehr laut.

- **Partnerarbeit** ist bei Gerätemangel gut geeignet: eine Person bedient die Maus, die andere liest die Aufgabe vor; nach jeder Aufgabe wird gewechselt.
- **Programme sichern** als festes Ritual am Stundenende (siehe oben).

Datenschutz

Wir haben den Quelltext von PBC+ Studio (Build 109) daraufhin durchgesehen, welche Daten die Seite verschickt:

- Es gibt **keine Anmeldung** und keine Konten. Programme und Bilder werden **nicht** an einen Server übertragen, sondern nur im Browser, als Datei auf dem Rechner oder per USB auf dem Gerät gespeichert.
- Es sind **keine Analyse- oder Werbedienste** eingebunden.
- Beim Laden ruft der Browser die Seite selbst und die oben genannten Bibliotheken von **cdnjs.cloudflare.com** und **cdn.jsdelivr.net** ab (MicroPython-WASM für den Emulator kommt von jsdelivr). Dabei sehen diese Anbieter – wie bei jedem Seitenabruf – technisch bedingt die IP-Adresse des Rechners bzw. der Schule.
- Die einzige Stelle, die Text an den Server von pbcstudio.de sendet, ist das **Kontaktformular** (Link „Kontakt“ unten auf der Seite). Es sendet nur, was dort eingetragen und mit „Senden“ abgeschickt wird. Weisen Sie die Klasse an, das Formular nicht zu benutzen.
- Was die beteiligten Webserver protokollieren, lässt sich aus dem Quelltext nicht ablesen. Den Anbieter nennt das **Impressum** auf der Seite. Klären Sie den Einsatz wie bei anderen Webangeboten nach den Regeln Ihrer Schule.

> **Checkliste vor der ersten Stunde**
>
> - pbcstudio.de auf einem Schülerrechner mit Schülerkonto öffnen, Emulator starten (lädt MicroPython), Ton prüfen.
> - Speicherort für `.pbcp`-Dateien und die Bilder aus `material` festlegen.
> - Wenn Geräte genutzt werden: Firmware aufspielen, Startmenü installieren, mit einem Testprogramm prüfen, Geräte und Datenkabel nummerieren.
> - Für AB 21 Geräte einplanen (kein Kippen im Emulator).

LEHRERMATERIAL · L 02

Unterrichtsreihe

Die Reihe führt Schülerinnen und Schüler der Klassen 5–7 in rund 30 Unterrichtsstunden von der ersten Anweisung bis zum selbst entwickelten Spiel. Dieses Heft zeigt den Gesamtplan, die Voraussetzungen und Varianten für weniger Zeit, wenige Geräte oder eine AG.

Aufbau der Reihe

Die 23 Arbeitsblätter folgen der Reihenfolge des Schülerbuchs und bauen aufeinander auf. Jedes Arbeitsblatt füllt eine Unterrichtsstunde (45 min); AB 20 und die Projekte AB 22 und AB 23 brauchen mehrere Stunden. Innerhalb eines Blatts steigen die Aufgaben von **nachbauen** über **verändern** bis **selbst erweitern**; ⭐ ist Pflicht, ⭐⭐ Zusatz, ⭐⭐⭐ für Schnelle.

Phase	Arbeitsblätter	Stunden	Leitfrage
I Los geht's	AB 01–02	2	Wie entsteht ein Programm, und wo erscheint etwas auf dem Bildschirm?
II Grundlagen	AB 03–10	8	Wie steuere ich Reihenfolge, Wiederholung, Entscheidung und Eingabe?
III Grafik und Bewegung	AB 11–17	7	Wie lasse ich Figuren zeichnen, sich bewegen und aufeinander reagieren?
IV Größere Programme	AB 18–21	5	Wie halte ich große Programme übersichtlich und speichere viele Werte?
V Projekte	AB 22–23	7–9	Wie plane, baue und teste ich ein eigenes Spiel?
	gesamt	**29–31**	zuzüglich Puffer- und Präsentationsstunden

Stoffverteilungsplan

> **Abkürzungen der Kompetenzspalte: A** Algorithmen/Sequenz · **V** Variablen · **Vz** Verzweigung · **S** Schleifen · **M** Modularisierung · **D** Daten/Listen · **E** Eingabe–Verarbeitung–Ausgabe · **K** Koordinaten und Geometrie · **T** Testen und Fehlersuche

AB	Titel	Std.	Inhalt / Lernziel	Kompetenzen	Kap.
01	PBC+ Studio kennenlernen	1	Oberfläche, Emulator, Testen und Stop, Speichern	A, T	1
02	Hallo PBC+!	1	Text an eine Position schreiben, Koordinatensystem	A, K	2
03	Ein Bild aus Formen	1	Haus/Gesicht aus Rechtecken, Kreisen, Linien; Farben	A, K	3
04	Die Ampel	1	Reihenfolge und Wartezeiten mit den Status-LEDs	A, E	4, 9
05	Muster mit Schleifen	1	Streifen, Treppen, Schachbrett mit Zählschleifen	S, K	5
06	Der springende Ball	1	Animation mit Variablen und Endlosschleife	V, S	5, 6
07	Der Würfel	1	Zufallszahl auf Tastendruck anzeigen	V, E	6, 8
08	Wenn – dann – sonst	1	Zahlenraten, Ampel auf Knopfdruck; Vergleiche, und/oder	Vz	7
09	Steuern mit dem Joystick	1	Punkt bewegen und am Rand stoppen; „solange“	Vz, S, E	8
10	Musik!	1	Töne, Pausen, eine eigene Melodie	A, S	9
11	Turtle 1: Vielecke	1	Quadrat, Dreieck, Stern – Winkel entdecken	S, K	10
12	Turtle 2: Spiralen und Muster	1	wachsende Längen, Farben wechseln	V, S, K	10
13	Turtle 3: Die Turtle sieht Farben	1	Turtle fährt bis zur Wand und dreht um	Vz, S	10
14	Mein Hintergrund	1	Canvas bemalen oder Bild laden und zeigen	A, K	11
15	Sprites in Bewegung	1	Sprite anlegen, steuern, gleiten	E, K	12

AB	Titel	Std.	Inhalt / Lernziel	Kompetenzen	Kap.
16	Kollision 1: Achtung, Hai!	1	Sprite berührt Sprite, Leben zählen	Vz, V	13
17	Kollision 2: Vorsicht, Lava!	1	Hintergrundfarben als Hindernis, Rand als Grenze	Vz, T	13
18	Ordnung mit Funktionen	1	unübersichtliches Programm in Funktionen aufteilen	M	14
19	Listen 1: Meine Bestenliste	1	Werte sammeln, zählen, alle anzeigen	D, S	15
20	Listen 2: Merk dir die Folge	2	„Simon sagt“: Folge in einer Liste, nachspielen	D, Vz, S	15
21	Kipp mich!	1	Kugel durch Neigen des Geräts bewegen	E, V	18
22	Projekt: Schatzjagd	3	Spiel aus Kapitel 20 schrittweise nachbauen und erweitern	M, D, T	20
23	Projekt: Mein eigenes Spiel	4–6	Planungsbogen, bauen, testen, vorstellen	alle	21

Die Buchkapitel 16 (Texte und Konsole), 17 (Mathe-Werkzeuge) und 19 (Reaktionstest) haben kein eigenes Arbeitsblatt. Sie eignen sich als Nachschlagewerk, für schnelle Schülerinnen und Schüler oder als zusätzliches Projekt (Reaktionstest, ca. 2 Stunden).

Voraussetzungen

Technik: sicherer Umgang mit Maus und Tastatur (ziehen, klicken, tippen), Dateien speichern und wiederfinden. Das wird in AB 01 geübt.

Mathematik: Die Reihe nutzt Inhalte, die in den Klassen 5–7 nach und nach dazukommen. Achten Sie auf den Stand Ihrer Klasse:

Inhalt	wird gebraucht in	Hilfe
Koordinaten ablesen und eintragen	ab AB 02	Koordinatenraster (L 07, Vorlage A); Bildschirm-y wächst nach **unten**
Natürliche Zahlen bis 1000, Rechnen +, −, ·	ab AB 05	–
Winkel in Grad, Drehungen	AB 11–13	Winkelscheibe (L 07, Vorlage B)
Negative Zahlen	AB 11–13 (Turtle-Koordinaten), Bewegungen	Turtle-Koordinatensystem (L 07, Vorlage C); vorher klären, falls noch nicht im

Inhalt	wird gebraucht in	Hilfe
	nach links/oben	Mathematikunterricht behandelt
Teilen mit Rest	vereinzelt (Muster, Spiele)	Erklärung im Buch, Kapitel 6.2

Lesen: Arbeitsblätter und Blöcke sind sprachlich einfach gehalten. Für schwächere Leserinnen und Leser hilft Partnerarbeit mit Vorlesen.

Kompetenzbezug

Die Reihe deckt zentrale Inhalte des Anfangsunterrichts in Informatik ab, wie sie in vielen Lehrplänen und Bildungsstandards für die Sekundarstufe I beschrieben werden. Die folgende Zuordnung ist allgemein gehalten; gleichen Sie sie mit dem für Ihre Schule gültigen Curriculum ab.

Bereich	Die Schülerinnen und Schüler …	Schwerpunkt
Algorithmen	lesen, verändern und entwerfen Abläufe als Folge von Anweisungen; erkennen, dass die Reihenfolge entscheidet	AB 01–04, 10, 14
Variablen	speichern Werte (Position, Punkte, Zeit), verändern und lesen sie; unterscheiden Zahl und Text	AB 06–07, 12, 16, 21
Verzweigung	formulieren Bedingungen mit Vergleichen und logischen Verknüpfungen und lassen das Programm darauf reagieren	AB 08–09, 13, 16–17
Schleifen	nutzen Zähl-, Bedingungs- und Endlosschleifen; erkennen Muster als Wiederholung	AB 05–06, 09, 11–12
Modularisierung	zerlegen Probleme in Teilprobleme, benennen sie als Funktionen und rufen sie auf, auch mit Parametern	AB 18, 22–23
Daten und Listen	speichern viele gleichartige Werte in einer Liste und verarbeiten sie mit Schleifen	AB 19–20, 22
Informatiksysteme	beschreiben Ein- und Ausgabe eines Geräts (Tasten, Sensor, Bildschirm, LEDs, Lautsprecher) nach dem EVA-Prinzip	AB 04, 07, 09–10, 21
Testen und Fehlersuche	testen Programme gezielt, deuten Fehlermeldungen und verbessern schrittweise	durchgehend, besonders AB 17, 22–23
Kommunizieren und Kooperieren	planen im Team, dokumentieren und präsentieren ein Produkt	AB 23

Varianten

Kurzreihe (etwa 10 Stunden)

Für einen Projekttag-Block, eine Vertretungsreihe oder als Einstieg in einer Klasse mit wenig Informatikzeit. Zwei Wege bieten sich an:

Std.	**Spiele-Kurzreihe**	**Turtle-Kurzreihe** (Mathematik-Bezug)
1	AB 01 PBC+ Studio kennenlernen	AB 01 PBC+ Studio kennenlernen
2	AB 02 Hallo PBC+!	AB 03 Ein Bild aus Formen
3	AB 05 Muster mit Schleifen	AB 05 Muster mit Schleifen
4	AB 06 Der springende Ball	AB 11 Turtle 1: Vielecke
5	AB 08 Wenn – dann – sonst	AB 12 Turtle 2: Spiralen und Muster
6	AB 09 Steuern mit dem Joystick	AB 06 Der springende Ball
7	AB 15 Sprites in Bewegung	AB 08 Wenn – dann – sonst
8	AB 16 Kollision 1: Achtung, Hai!	AB 13 Turtle 3: Die Turtle sieht Farben
9–10	Mini-Projekt: AB 16 zu einem eigenen Spiel ausbauen (Planungsbogen aus L 07 in Kurzform)	Mini-Projekt: Musterbild oder Turtle-Labyrinth, Ausstellung

AB 02 und AB 03 lassen sich in der Spiele-Kurzreihe zu einer Stunde zusammenfassen, wenn die Klasse schnell ist; AB 07 (Würfel) ist dann ein guter Ersatz. Die Kurzreihen verzichten auf Funktionen und Listen.

Langreihe (gesamte Reihe, ca. 30 Stunden)

Alle Arbeitsblätter in der angegebenen Reihenfolge, z. B. ein Halbjahr mit zwei Wochenstunden. Planen Sie 2–3 Pufferstunden ein: nach AB 10 (Wiederholung, kleines Zwischenprojekt), nach AB 17 (Spiel aus AB 16/17 ausbauen) und vor den Präsentationen in AB 23. Kapitel 19 (Reaktionstest) eignet sich als zusätzliches Zwischenprojekt vor AB 22.

Mit Geräten, mit wenigen Geräten, ohne Geräte

	Empfehlung
Ein Gerät pro Person oder Paar	Ab AB 01 Aufgabe 6 mit dem Gerät arbeiten; Emulator weiter zum schnellen Testen. Startmenü installieren, damit fertige Programme auf dem Gerät bleiben (L 01).

	Empfehlung
Wenige Geräte	Entwicklung im Emulator; eine oder zwei „Gerätestationen“, an denen fertige Stände ausprobiert werden. Besonders lohnend: AB 04, 09, 10, 15–17, 21–23.
Ohne Geräte	Alle Arbeitsblätter außer AB 21 funktionieren vollständig im Emulator. AB 21 kann entfallen, als Vorführung mit einem Lehrergerät per Dokumentenkamera laufen oder durch eine Joystick-Variante ersetzt werden (die Kugel mit dem Joystick steuern statt durch Kippen).

Arbeitsgemeinschaft

In einer AG mit gemischten Jahrgängen und freiwilliger Teilnahme lohnt ein projektorientierter Ablauf: die Grundlagen (AB 01–10) zügig, je Treffen zwei Arbeitsblätter mit den ⭐-Aufgaben; danach Grafik und Kollision (AB 11–17) und früh ein eigenes Spielprojekt (AB 23), das bis zum Halbjahresende wächst. Funktionen und Listen (AB 18–20) werden dann eingeführt, wenn das eigene Projekt sie braucht. Ein festes Ritual am Ende jedes Treffens – „Zeigt her, was ihr gebaut habt“ – hält die Motivation hoch. Ein Abschluss vor Publikum (Tag der offenen Tür, Spieleabend) ist ein schönes Ziel.

Hinweise zur Durchführung

- **Sichern als Ritual:** Am Ende jeder Stunde **Dateien → Auf den Computer…** mit sprechendem Dateinamen. Viele Arbeitsblätter bauen auf der Vorstunde auf.
- **Lösungsdateien** (`.pbcp`) zu jedem Arbeitsblatt gibt es auf Anfrage. Sie lassen sich mit **Dateien → Vom Computer…** öffnen und als Startpunkt verteilen, wenn jemand gefehlt hat.
- **Beamer:** Das Vorführen im Emulator vor der Klasse eignet sich gut für Einstieg und Sicherung.
- **Fehler sind Material:** Lassen Sie Fehlermeldungen im Terminal vorlesen und gemeinsam deuten. Typische Fehler und Hilfen stehen in jedem Lehrermaterial zum Arbeitsblatt.
- **Bewertung** der Projekte AB 22 und AB 23: L 06.

LEHRERMATERIAL · AB 01

PBC+ Studio kennenlernen

Überblick

Lernziel	Die Schülerinnen und Schüler kennen die Bereiche von PBC+ Studio, testen ein Programm im Emulator, ändern Werte in Blöcken, holen und löschen Blöcke und speichern ihr Programm.
Dauer	45 Minuten (Aufgabe 1–4 Pflicht, 5–6 Differenzierung)
Voraussetzungen	keine
Material	Rechner mit Chrome oder Edge, PBC+ Studio (pbcstudio.de, Build 109 oder neuer). Optional: PicoBoys mit USB-Kabel für Aufgabe 6. Lösungsdateien auf Anfrage
Buch	Kapitel 1

Vorbereitung: PBC+ Studio einmal auf einem Schülerrechner öffnen und prüfen, ob der Emulator startet (er lädt beim ersten Mal MicroPython aus dem Internet nach). Für Töne im Emulator Kopfhörer bereitlegen oder die Lautstärke der Rechner abstimmen.

> PBC+ Studio speichert das zuletzt bearbeitete Programm im Browser. An geteilten Schulrechnern findet die nächste Klasse also das Programm der vorherigen vor. **Dateien → Neu** setzt die Arbeitsfläche auf das Startprogramm zurück.

Stundenverlauf

Zeit	Phase	Inhalt	Sozialform
5 min	Einstieg	PicoBoy zeigen (oder Emulator per Beamer). „Was kann so ein Gerät? Was braucht man, um es zu programmieren?“	Plenum
10 min	Aufgabe 1	Bereiche der Oberfläche zuordnen, danach gemeinsam vergleichen	Einzel, Plenum
10 min	Aufgabe 2–3	Startprogramm testen und verändern; Block löschen und neuen Block einfügen	Einzel/Partner
10 min	Aufgabe 4	Speichern, „Neu“, wieder öffnen	Einzel
5 min	Aufgabe 5/6	Ton ergänzen; wenn Geräte da sind: auf dem PicoBoy testen	Einzel/Partner

Zeit	Phase	Inhalt	Sozialform
5 min	Sicherung	„Was passiert bei ▶ Testen?“ – Reihenfolge von oben nach unten, nur Blöcke unter „Beim Start“	Plenum

Typische Fehler

Beobachtung	Ursache	Hilfe
Nach ▶ Testen erscheint nur die Meldung „Erst verbinden.“	Ziel steht auf **PBC+**, es ist aber kein Gerät verbunden.	Ziel **Emulator** wählen.
Ein Block wirkt nicht.	Er hängt nicht unter „Beim Start“, sondern liegt frei auf der Arbeitsfläche.	Block an die Kette anhängen, bis er einschnappt.
Beim Löschen ist auch der Text-Block weg.	Der Block wurde auf die Blockleiste gezogen – die Blöcke darunter kommen mit.	Mit Strg+Z rückgängig machen; zum Löschen einzelner Blöcke **Entf** nutzen.
Der Text ist abgeschnitten.	Langer Text bei x = 74.	x verkleinern.
Kein Ton zu hören.	Lautsprecher oder Kopfhörer stumm.	Lautstärke am Rechner prüfen, erneut testen.
„Verbinden“ zeigt kein Gerät.	Falscher Browser (z. B. Firefox, Safari) oder Kabel nur zum Laden.	Chrome oder Edge nutzen; anderes USB-Kabel probieren.
Nach „Neu“ ist das Programm weg.	Nicht gespeichert.	Vorher **Dateien → Auf den Computer…**

LEHRERMATERIAL · AB 02

Hallo PBC+!

Überblick

Lernziel	Die Schülerinnen und Schüler schreiben Texte an vorgegebene Bildschirmpositionen, lesen Koordinaten im Bildschirm-Koordinatensystem (Ursprung oben links, y nach unten) ab, nutzen die drei Textgrößen und gliedern ein Programm mit Kommentaren.
Dauer	45 Minuten (Aufgabe 1–3 Pflicht, 4–5 Differenzierung)
Voraussetzungen	AB 01 (Blöcke holen, Werte ändern, Testen)
Material	PBC+ Studio (Build 109 oder neuer), Lösungsdateien auf Anfrage. Optional: Kopiervorlage Koordinatenraster (L 07)
Buch	Kapitel 2

Vorbereitung: Im Mathematikunterricht der Klassen 5/6 zeigt die y-Achse nach oben. Planen Sie Zeit ein, um den Unterschied ausdrücklich zu besprechen – er ist die häufigste Fehlerquelle der Stunde.

Stundenverlauf

Zeit	Phase	Inhalt	Sozialform
5 min	Einstieg	Emulator per Beamer: „Hallo PBC+!“ erscheint. Frage: „Woher weiß der Computer, wo der Text hin soll?“	Plenum
8 min	Aufgabe 1	Programm nachbauen, x und y auf 0 setzen, Wirkung beschreiben	Einzel
7 min	Sicherung 1	Koordinatensystem an der Tafel: (0, 0) oben links, y nach unten, Bildschirm 240 × 280	Plenum
8 min	Aufgabe 2	Koordinaten ablesen, ein Ergebnis im Emulator prüfen	Einzel/Partner
10 min	Aufgabe 3–5	Drei Textgrößen; schnelle SuS: Visitenkarte mit Kommentaren, vier Ecken	Einzel
7 min	Sicherung 2	„Nachgedacht“ (Umlaute) und Aufgabe 5 (runde Ecken) besprechen	Plenum

Typische Fehler

Beobachtung	Ursache	Hilfe
Der Text steht ganz oben links in der Ecke.	Ein frisch geholter Text-Block hat überall 0 eingetragen.	x und y ändern.
Der Text erscheint nicht oder nur teilweise.	x oder y zu groß (z. B. y = 300), der Text liegt außerhalb des Bildschirms. Eine Fehlermeldung gibt es nicht.	Zahlen mit dem Raster aus Aufgabe 2 vergleichen: x bis 239, y bis 279.
Text steht unten statt oben (oder umgekehrt).	y wird wie im Matheheft nach oben gezählt.	Merksatz: „y zählt nach unten – wie Zeilen in einem Heft.“
Zwei Texte überlappen sich.	y-Abstand kleiner als die Texthöhe (16/24/32).	Abstand vergrößern.
Statt Buchstaben erscheinen „?“.	Umlaute oder ß im Text.	ae, oe, ue, ss schreiben.
Der Kommentar „wirkt nicht“.	Erwartung, dass der Kommentar etwas anzeigt.	Kommentare sind nur Notizen für Menschen; das Programm überspringt sie.
Ein Block wird nicht ausgeführt.	Er hängt nicht unter „Beim Start“.	Einschnappen lassen.

Differenzierung

- **Langsamere SuS:** Aufgabe 2 mit der Kopiervorlage Koordinatenraster (L 07) und Lineal bearbeiten; Aufgabe 3 nur mit zwei Textgrößen.
- **Schnelle SuS:** Aufgabe 5; zusätzlich ein Wort genau in die Bildschirmmitte setzen (Buch, Kapitel 2, Aufgabe 5).
- **Gerät:** Wer einen PicoBoy hat, testet Aufgabe 5 auf dem Gerät – die runden Ecken sieht man dort genauso.

Ein Bild aus Formen

Überblick

Lernziel	Die Schülerinnen und Schüler zeichnen ein Bild aus Rechtecken, Linien, Kreisen und Ellipsen nach Koordinatenangaben, mischen Farben aus Rot, Grün und Blau und erkennen, dass später Gezeichnetes Früheres überdeckt.
Dauer	45 Minuten (Aufgabe 1–2 Pflicht, 3–5 Differenzierung)
Voraussetzungen	AB 02 (Koordinatensystem, Werte in Blöcken ändern)
Material	PBC+ Studio (Build 109 oder neuer), Kästchenpapier für Aufgabe 5, Lösungsdateien auf Anfrage
Buch	Kapitel 3.1–3.4

Vorbereitung: Ein fertiges Haus (Lösung A3) im Emulator bereithalten. Der Bauplan in Aufgabe 1 ist bewusst eine Tabelle und keine Blockvorlage: Die SuS übertragen Maße selbst in die Blöcke.

Stundenverlauf

Zeit	Phase	Inhalt	Sozialform
5 min	Einstieg	Fertiges Haus zeigen: „Aus welchen Formen besteht das Bild?“ Formen an der Tafel sammeln.	Plenum
5 min	Erarbeitung	Rechteck-Block vorführen: Ecke oben links, Breite, Höhe; „Umriss“ / „gefüllt“	Plenum
12 min	Aufgabe 1	Haus nach Bauplan; Kopieren per Rechtsklick	Einzel/Partner
8 min	Aufgabe 2	Dach und Sonne; Unterschied Ecke (Rechteck) – Mitte (Kreis)	Einzel/Partner
10 min	Aufgabe 3–5	Farben mischen, Smiley, eigenes Bild	Einzel
5 min	Sicherung	Frage aus A3: Reihenfolge beim Zeichnen; Lichtmischung (Rot + Grün = Gelb)	Plenum

Typische Fehler

Beobachtung	Ursache	Hilfe
Das Rechteck ist nur ein Rahmen.	Ein frisch geholter Rechteck-Block steht auf **Umriss**.	Ganz vorne im Block „gefüllt“ wählen.
Ein Rechteck ist nicht zu sehen.	b oder h ist noch 0 (Startwert), oder die Farbe ist schwarz auf schwarzem Grund.	Werte prüfen.
Kreis viel zu groß.	Durchmesser statt Radius eingetragen.	Radius = halbe Breite.
Ellipse halb so groß wie erwartet bzw. Kreis doppelt so groß wie erwartet.	Bei der Ellipse ist die **ganze** Breite gemeint, beim Kreis der Radius.	Buch, Kapitel 3.2, Kasten „Mitte, nicht Ecke!“
Kreis liegt verschoben.	x, y als Ecke statt als Mitte verstanden.	Bei Kreis und Ellipse ist x, y die Mitte.
Fenster oder Tür verschwinden.	Die Mauer wurde **nach** den Fenstern gezeichnet und übermalt sie.	Reihenfolge: hinten zuerst, vorne zuletzt.
Nur Himmel zu sehen.	„Bildschirm füllen“ steht unten.	Ganz nach oben schieben.
Farbe(r, g, b) bleibt schwarz.	Alle drei Werte noch 0 (Startwert).	Werte zwischen 0 und 255 eintragen.

Differenzierung

- **Langsamere SuS:** Aufgabe 1 zu zweit (einer liest die Tabelle vor, einer baut); nur drei Rechtecke (Wiese, Mauer, Tür).
- **Schnelle SuS:** Aufgabe 5; das Dach mit einer Zählschleife ausfüllen (Ausblick auf Kapitel 5, Aufgabe 5).
- **Fächerverbindend (Physik/Kunst):** Lichtmischung mit dem Block Farbe(r, g, b) erkunden – Rot + Grün = Gelb, alle drei = Weiß.

LEHRERMATERIAL · AB 04

Die Ampel

Überblick

Lernziel	Die Schülerinnen und Schüler steuern die drei Status-LEDs, setzen Wartezeiten in Sekunden und Millisekunden ein und erfahren, dass die Reihenfolge der Blöcke den Ablauf bestimmt (Sequenz).
Dauer	45 Minuten (Aufgabe 1–2 Pflicht, 3–5 Differenzierung)
Voraussetzungen	AB 02 (Text-Blöcke), sicherer Umgang mit Holen und Einhängen von Blöcken
Material	PBC+ Studio (Build 109 oder neuer); optional PicoBoys. Lösungsdateien auf Anfrage
Buch	Kapitel 4 (Reihenfolge und Zeit); die LED-Blöcke ausführlich in Kapitel 9.1

Vorbereitung: Die LEDs sitzen im Emulator links neben dem Bildschirm und sind klein. Zeigen Sie sie per Beamer. Mit echten PicoBoys ist die Stunde besonders anschaulich – die LEDs leuchten dort hell.

> Im Emulator bleiben die LEDs so, wie das Programm sie zuletzt gesetzt hat – auch nach ■ **Stop** und beim nächsten Start. Deshalb beginnt die Lösung von Aufgabe 2 mit „alle Status-LEDs aus“. Machen Sie das zur Regel.

Stundenverlauf

Zeit	Phase	Inhalt	Sozialform
5 min	Einstieg	Bild einer Fußgängerampel. „Was macht die Ampel nacheinander? Wie lange?“ Phasen an der Tafel sammeln.	Plenum
8 min	Aufgabe 1	LED- und Warte-Blöcke kennenlernen; Wirkung von „LED aus“ entdecken	Einzel
15 min	Aufgabe 2	Ampelphasen nach Tabelle programmieren; Kommentare je Phase	Partner
10 min	Aufgabe 3–5	Fußgängeranzeige, Zeit messen, Nachtbetrieb	Einzel/Partner
7 min	Sicherung	Ein Programm vorführen. Merksätze: „Der Computer macht genau das, was dasteht – Schritt für Schritt.“ „Ohne Warten geht alles sofort.“ Ausblick: Wiederholungen ersparen das Kopieren (AB 05).	Plenum

Typische Fehler

Beobachtung	Ursache	Hilfe
Alle Phasen „rauschen“ vorbei, am Ende leuchtet nur die letzte Farbe.	Warte-Block fehlt oder steht auf 0 (Startwert eines neuen Warte-Blocks).	Nach jeder Phase einen Warte-Block mit Zahl.
Zwei oder drei LEDs leuchten gleichzeitig.	Eine LED wurde beim Phasenwechsel nicht ausgeschaltet.	Für jede Phase prüfen: Was muss an, was aus?
Die Ampel wartet viel zu lange (oder nur kurz).	„warte … Sekunden“ und „warte … ms“ verwechselt (z. B. warte 3 ms oder warte 3000 Sekunden).	Einheit im Block lesen; 1000 ms = 1 s.
Schon beim Start leuchtet eine LED.	Die LEDs stehen noch vom vorigen Testlauf.	„alle Status-LEDs aus“ an den Anfang.
Im LED-Block lässt sich keine Zahl eintragen.	Farbe und an/aus werden über die Auswahllisten (▾) gewählt.	Auf „rot ▾“ bzw. „an ▾“ klicken.
Text STOP/GEHEN überlagert sich.	„Bildschirm löschen“ vor dem neuen Wort fehlt.	Erst löschen, dann schreiben.
Die angezeigte Zeit ist 0 oder sehr klein.	Der Zeit-Block steht am Anfang statt am Ende des Programms.	Ganz unten anhängen.

Differenzierung

- **Langsamere SuS:** Aufgabe 2 nur mit drei Phasen (Rot – Grün – Gelb). Tabelle mit Spalte „an“ / „aus“ an der Tafel vorgeben.
- **Schnelle SuS:** Aufgabe 5; eine Fußgängerampel mit Rot/Grün und gleichzeitiger Autoampel auf dem Bildschirm (Kreise aus AB 03).
- **Mit Geräten:** Programm mit **Dateien → Auf den PBC+…** als **Python-Programm** speichern und im Startmenü starten (Buch, Kapitel 1.6).

LEHRERMATERIAL · AB 05

Muster mit Schleifen

Überblick

Lernziel	Die Schülerinnen und Schüler wiederholen Blöcke mit „wiederhole … -mal“, nutzen die Zählvariable der Schleife „für i von … bis… in Schritten von …“ als Koordinate und verschachteln zwei Zählschleifen.
Dauer	45 Minuten (Aufgabe 1–3 Pflicht, 4–5 Differenzierung)
Voraussetzungen	AB 03 (Rechteck, Koordinaten), AB 04 (warten)
Material	PBC+ Studio (Build 109 oder neuer), Lösungsdateien auf Anfrage
Buch	Kapitel 5.1–5.2

Vorbereitung: Den Unterschied zwischen dem Buchstaben „i“ in einem Textfeld und dem Variablenblock `i` (Kategorie **Variablen**, erscheint, sobald eine Zählschleife auf der Arbeitsfläche liegt) einmal vorführen. Gut eignet sich dafür das Programm `k05_zahlen.pbcp` aus dem Buch: Es schreibt die Werte von `i` auf den Bildschirm.

Stundenverlauf

Zeit	Phase	Inhalt	Sozialform
5 min	Einstieg	Rückblick AB 04, Aufgabe 5: 12 Blöcke für dreimal Blinken. „Geht das kürzer?“	Plenum
7 min	Aufgabe 1	N-mal-Schleife: Blinklicht	Einzel
8 min	Erarbeitung	Zählschleife vorführen (`k05_zahlen.pbcp`): Welche Werte nimmt `i` an? Tabelle an der Tafel	Plenum
10 min	Aufgabe 2–3	Streifen und Treppe	Einzel/Partner
10 min	Aufgabe 4–5	Schachbrett (verschachtelte Schleifen), eigenes Muster	Partner
5 min	Sicherung	„Nachgedacht“: Schleifenende; Merksatz „Die Schleife hört auf, bevor i größer als ‚bis‘ wird.“	Plenum

Typische Fehler

Beobachtung	Ursache	Hilfe
Nichts passiert, kein Fehler.	„wiederhole … -mal“ steht noch auf 0 (Startwert).	Anzahl eintragen.
Das Blinken ist nicht zu sehen.	Warte-Blöcke fehlen oder stehen auf 0.	In der Schleife zweimal warten.
Nur ein Streifen, immer an derselben Stelle.	Bei x steht eine feste Zahl statt des Blocks `i`.	Block `i` aus **Variablen** einsetzen.
Auf dem Bildschirm erscheint der Buchstabe „i“.	„i“ in ein Textfeld getippt statt den Variablenblock zu nutzen.	Variablenblock verwenden.
Die Blöcke laufen nur einmal.	Sie hängen **unter** statt **in** der Schleife.	In die Lücke („mache“) ziehen.
Schachbrett: nur die erste Reihe wird gezeichnet.	Beim Rechteck steht bei y noch 0 statt `j`.	y = `j`.
Schachbrett: alle Quadrate in einer Spalte.	`i` und `j` vertauscht oder beide Felder mit `j` belegt.	x = `i`, y = `j`.

Differenzierung

- **Langsamere SuS:** Aufgabe 4 entfällt; stattdessen Ringe mit „Kreis Umriss, Radius `i`“ (Buch, Kapitel 5.2).
- **Schnelle SuS:** Schachbrett mit kleineren Feldern (Schrittweite 40, Quadrate 20) oder das Dach des Hauses mit einer Schleife füllen (Buch, Kapitel 5, Aufgabe 5).
- **Ausblick:** Animation mit „unendlich wiederholen“ und warum es flackert – Buch, Kapitel 5.3–5.4 (Vorbereitung auf AB 06).

Der springende Ball

Überblick

Lernziel	Die Schülerinnen und Schüler legen Variablen an, verändern sie in einer Endlosschleife („setze y auf y + dy“) und erzeugen damit eine flimmerfreie Animation (alte Stelle übermalen, neu zeichnen).
Dauer	45 Minuten (Aufgabe 1–3 Pflicht, 4–5 Differenzierung)
Voraussetzungen	Zeichnen mit Kreis (AB 03), Schleifen (AB 05)
Material	PBC+ Studio (Build 109 oder neuer), Lösungsdateien auf Anfrage
Buch	Kapitel 6.1 und 6.2, Kapitel 8.4 (Übermalen statt Löschen)

Vorbereitung: Keine besondere. Der „falls“-Block in Aufgabe 3 wird erst in AB 08 richtig eingeführt; hier bauen die Kinder ihn nach Vorlage nach. Es lohnt sich, das Flackern aus Aufgabe 2 einmal per Beamer zu zeigen.

Stundenverlauf

Zeit	Phase	Inhalt	Sozialform
5 min	Einstieg	Lösung A4 im Emulator zeigen. „Woher weiß der Computer, wo der Ball gerade ist?“ → Er muss sich die Position merken: Variable = Schachtel mit Namen.	Plenum
10 min	Vorbereitung, Aufgabe 1	Variable `y` anlegen, Vorlage nachbauen	Einzel/Partner
8 min	Aufgabe 2	Experiment: ohne Übermalen (Spur), mit „Bildschirm löschen“ (Flackern)	Partner
5 min	Zwischensicherung	Warum flackert es? Regel: alte Stelle übermalen, sofort neu zeichnen	Plenum
12 min	Aufgabe 3–5	Abprallen mit `dy`, schräg, Ton	Einzel/Partner
5 min	Sicherung	„setze y auf y + dy“ laut lesen: „Mach y um dy größer“. Was bewirkt „0 − dy“?	Plenum

Typische Fehler

Beobachtung	Ursache	Hilfe
Der Ball bewegt sich nicht.	„setze y auf …“ steht vor der Schleife statt darin, oder es steht „setze y auf 3“ statt „y + 3“.	Block in die Schleife ziehen; Rechen-Block „y + 3“ einsetzen.
Der Ball zieht eine Spur.	Der schwarze Kreis fehlt, hat eine andere Position (z. B. 120 statt `y`) oder einen kleineren Radius.	Beide Kreise müssen dieselbe Mitte und denselben Radius haben.
Der Ball prallt unten nicht ab, sondern fällt aus dem Bild.	„setze dy auf 3“ steht in der Schleife und macht das Umkehren in jeder Runde wieder rückgängig.	Startwerte gehören vor die Schleife (über den Dialog landen sie automatisch unter „Beim Start“).
Der Ball zittert gleich am Anfang oben auf der Stelle.	Im „falls“ steht `y < 265` statt `y > 265`: Die Richtung kehrt sich in jeder Runde um.	Vergleichszeichen im Feld ▾ prüfen.
Unter „Beim Start“ stehen mehrere „setze y auf …“.	Die Variable wurde mehrmals mit Startwert angelegt bzw. ein Block zusätzlich geholt.	Doppelte Blöcke löschen.

Differenzierung

- **Schneller fertig:** Zweiter Ball mit `x2`, `y2`, `dx2`, `dy2` – oder ein Ball, der bei jedem Aufprall eine Zufallsfarbe bekommt (Farbe in einer Variable speichern, siehe Buch Kapitel 7, Ausprobieren).
- **Mehr Unterstützung:** `ab06_a1.pbcp` als Startdatei geben, dann direkt mit Aufgabe 2 beginnen.

Der Würfel

Überblick

Lernziel	Die Schülerinnen und Schüler erzeugen mit „Zufallszahl zwischen … und …“ Würfelzahlen, speichern sie in einer Variable und reagieren mit „wurde A seit der letzten Abfrage gedrückt?“ auf jeden einzelnen Tastendruck.
Dauer	45 Minuten (Aufgabe 1–3 Pflicht, 4–6 Differenzierung)
Voraussetzungen	Variablen anlegen (AB 06), unendlich wiederholen (AB 05/06)
Material	PBC+ Studio (Build 109 oder neuer), Lösungsdateien auf Anfrage
Buch	Kapitel 6.3 (Zufallszahlen), Kapitel 8.2 („wurde … gedrückt?“)

Vorbereitung: Für den Einstieg einen echten Würfel mitbringen. Den „falls“-Block lernen die Kinder hier nur als Muster kennen („falls Taste gedrückt, dann …“); Vergleiche folgen in AB 08.

Stundenverlauf

Zeit	Phase	Inhalt	Sozialform
5 min	Einstieg	Echter Würfel: Welche Zahlen sind möglich? Kann ein Computer würfeln?	Plenum
8 min	Aufgabe 1	Zufallszahl anzeigen, mehrmals testen, Bereich 0–6 ausprobieren	Einzel
12 min	Aufgabe 2–3	Würfeln mit A, Text und Ton	Einzel/Partner
5 min	Aufgabe 4	„wird“ statt „wurde“ – Beobachtung austauschen	Partner
10 min	Aufgabe 5–6	Würfe zählen, rollender Würfel	Einzel/Partner
5 min	Sicherung	Unterschied „wird … gedrückt?“ / „wurde … gedrückt?“ an der Tafel (Bild aus dem Buch, Kapitel 8.2)	Plenum

Typische Fehler

Beobachtung	Ursache	Hilfe
Die Zahlen stehen übereinander und sind	Das schwarze Rechteck fehlt oder steht **nach** dem Text.	Erst übermalen, dann schreiben.

Beobachtung	Ursache	Hilfe
unlesbar.		
In Aufgabe 3 bleiben Reste des alten Textes stehen.	Das Rechteck ist noch so schmal wie in Aufgabe 2.	x = 0, b = 240.
Es erscheint „Du hast eine4“.	Leerzeichen am Ende des Textes vergessen.	„Du hast eine “ mit Leerzeichen.
Es wird immer dieselbe Zahl gewürfelt.	„setze zahl auf Zufallszahl …“ steht vor der Schleife statt im „falls“.	Block in das „falls“ ziehen.
Beim Drücken von A passiert nichts.	Im Feld ▾ ist eine andere Taste (z. B. HOCH) gewählt, oder der „falls“-Block hängt nicht in der Schleife.	Feld auf A stellen; Block einrasten lassen.
„W?rfel“ auf dem Bildschirm	Die Bildschirmschrift kennt keine Umlaute.	„Wuerfel“ schreiben.

Differenzierung

- **Schneller fertig:** Zwei Würfel nebeneinander und ihre Summe anzeigen; bei einem Pasch einen besonderen Ton spielen (dafür braucht es „falls“ mit Vergleich, Vorgriff auf AB 08).
- **Mehr Unterstützung:** `ab07_a2.pbcp` als Startdatei geben; Aufgabe 1 nur besprechen.

LEHRERMATERIAL · AB 08

Wenn – dann – sonst

Überblick

Lernziel	Die Schülerinnen und Schüler verwenden „falls / sonst“, Vergleiche (= < > ≤) und die Verknüpfungen „und“ / „oder“, auch ineinander geschachtelt.
Dauer	45 Minuten (Aufgabe 1, 3 und 4 Pflicht, 2 und 5 Differenzierung)
Voraussetzungen	Variablen (AB 06), „wird / wurde … gedrückt?“ (AB 07)
Material	PBC+ Studio (Build 109 oder neuer), Lösungsdateien auf Anfrage
Buch	Kapitel 7

Vorbereitung: Die Vorlage für das Zahlenraten ist recht lang. Für langsamere Gruppen `ab08_a4_vorlage.pbcp` bereitstellen. Für Aufgabe 2 sitzen am besten zwei Kinder an einem Rechner (eins drückt A, eins B).

Stundenverlauf

Zeit	Phase	Inhalt	Sozialform
5 min	Einstieg	„Wenn es regnet, nehme ich den Schirm, sonst die Sonnenbrille.“ Weitere Wenn-dann-sonst-Sätze sammeln.	Plenum
8 min	Aufgabe 1	Fußgängerampel	Einzel/Partner
7 min	Aufgabe 2–3	und / oder ausprobieren, Wahr-oder-falsch-Tabelle	Partner
5 min	Zwischensicherung	Tabelle vergleichen; „und“: beide, „oder“: mindestens eins	Plenum
15 min	Aufgabe 4–5	Zahlenraten	Einzel/Partner
5 min	Sicherung	Ein Kind spielt das Zahlenraten vor; warum drei Möglichkeiten zwei „falls / sonst“ brauchen	Plenum

Typische Fehler

Beobachtung	Ursache	Hilfe
Die Ampel reagiert nicht auf A.	Der „falls / sonst“-Block steht unter der Schleife statt darin.	Block in die Schleife ziehen.
Beide LEDs leuchten gleichzeitig.	In einem Zweig fehlt das Ausschalten der anderen LED.	In jedem Zweig eine LED an, die andere aus.
Beim Zahlenraten springt der Tipp bei einem Druck um mehrere Zahlen.	„wird HOCH gedrückt?“ statt „wurde HOCH seit der letzten Abfrage gedrückt?“.	Block tauschen (siehe AB 07, Aufgabe 4).
Es erscheint immer „zu gross“, auch bei der richtigen Zahl.	Reihenfolge vertauscht: zuerst „tipp < geheim“, dann sofort sonst „zu gross“ – der Fall „gleich“ fehlt.	Zuerst auf „=“ prüfen, im sonst-Teil dann auf „<“.
Die Meldungen überlagern sich („zu kleinig!“).	Das schwarze Rechteck vor der Meldung fehlt.	Rechteck x 0, y 170, b 240, h 24 vor jeden Text.

Differenzierung

- **Schneller fertig:** Zahlen von 1 bis 100 raten; LINKS/RECHTS ändern den Tipp um 10. Bei „Richtig!“ die grüne LED einschalten.
- **Mehr Unterstützung:** `ab08_a4_vorlage.pbcp` als Startdatei; das geschachtelte „falls / sonst“ an der Tafel gemeinsam aufbauen.

Steuern mit dem Joystick

Überblick

Lernziel	Die Schülerinnen und Schüler steuern einen Punkt mit „wird … gedrückt?“ in vier Richtungen, begrenzen die Bewegung mit „und“-Bedingungen und beenden ein Spiel mit „wiederhole solange“.
Dauer	45 Minuten (Aufgabe 1–2 Pflicht, 3–5 Differenzierung)
Voraussetzungen	Variablen (AB 06), „falls“ und Vergleiche (AB 08)
Material	PBC+ Studio (Build 109 oder neuer), Lösungsdateien auf Anfrage
Buch	Kapitel 8.1 und 8.4, Kapitel 7.4 (solange)

Vorbereitung: Auf dem echten Gerät macht diese Stunde besonders viel Spaß, weil der Joystick „echt“ ist. Im Emulator steuern die Pfeiltasten.

Stundenverlauf

Zeit	Phase	Inhalt	Sozialform
5 min	Einstieg	Wie bewegt sich eine Spielfigur? Koordinaten wiederholen: x nach rechts, y nach **unten**.	Plenum
12 min	Aufgabe 1	Vorlage nachbauen, vier Richtungen	Einzel/Partner
8 min	Aufgabe 2	Halt am Rand mit „und“	Einzel/Partner
15 min	Aufgabe 3–5	Zielfeld mit „solange“, Turbo, Zeit	Einzel/Partner
5 min	Sicherung	Warum „oder“ in der solange-Bedingung? Wann ist man im Ziel?	Plenum

Typische Fehler

Beobachtung	Ursache	Hilfe
Der Punkt zieht eine Spur.	Der schwarze Kreis fehlt oder hat eine andere Mitte als der gelbe.	Beide Kreise: Mitte `x`, `y`, Radius 6.
Der Punkt flackert stark.	Statt des schwarzen Kreises wurde „Bildschirm löschen“	Nur die alte Stelle übermalen.

Beobachtung	Ursache	Hilfe
	verwendet.	
HOCH bewegt den Punkt nach unten.	`y + 3` statt `y - 3`; y zählt auf dem Bildschirm nach unten.	Rechenzeichen im Feld ▾ ändern.
Der Punkt bewegt sich nur einmal ein Stück.	Die „falls“-Blöcke stehen vor der Schleife.	In die Schleife ziehen.
Nach dem Tausch gegen „solange“ passiert nichts mehr.	Die Blöcke aus „unendlich wiederholen“ wurden nicht in die neue Schleife gezogen.	Den obersten Block im Inneren der alten Schleife greifen – die Blöcke darunter kommen mit – und in die neue Schleife ziehen.
Das Spiel endet schon, wenn der Punkt nur ganz unten **oder** nur ganz rechts ist.	„und“ statt „oder“ in der solange-Bedingung.	„oder“ wählen, Begründung siehe Lösung zu Aufgabe 3.

Differenzierung

- **Schneller fertig:** Eine rote „Lava“-Wand malen; berührt der Punkt sie (Bedingung mit x und y), geht es zurück zum Start.
- **Mehr Unterstützung:** `ab09_a1.pbcp` als Startdatei für Aufgabe 2.

LEHRERMATERIAL · AB 10

Musik!

Überblick

Lernziel	Die Schülerinnen und Schüler spielen Töne mit „Ton … Hz für … ms“, verstehen, dass dieser Block nicht wartet, bauen Melodien aus Ton und Pause und verwenden „Dauerton“ und „Ton stoppen“.
Dauer	45 Minuten (Aufgabe 1–3 Pflicht, 4–6 Differenzierung)
Voraussetzungen	Schleifen (AB 05); für Aufgabe 5: „wird … gedrückt?“ und „falls / sonst“ (AB 08/09)
Material	PBC+ Studio (Build 109 oder neuer), **Kopfhörer** (sonst wird es laut), Lösungsdateien auf Anfrage; ganzes Lied: `k09_loesung3.pbcp`
Buch	Kapitel 9.3 und 9.4

Vorbereitung: Lautstärke der Rechner prüfen und Kopfhörer austeilen. Im Emulator kommen die Töne aus dem Browser. Manche Browser spielen erst Ton ab, nachdem man einmal auf die Seite geklickt hat – das ist mit dem Klick auf ▶ **Testen** erledigt.

Stundenverlauf

Zeit	Phase	Inhalt	Sozialform
5 min	Einstieg	Lehrkraft spielt „Hänschen klein“ (Lösung A3) vor. „Wie macht der PBC+ das?“ – Tonhöhe und Tondauer	Plenum
7 min	Aufgabe 1	Hoch und tief, Hertz entdecken	Einzel
10 min	Aufgabe 2	Tonleiter; Experiment ohne „warte“	Einzel/Partner
5 min	Zwischensicherung	Der Ton-Block wartet nicht → nach jedem Ton ein „warte“	Plenum
13 min	Aufgabe 3–6	Hänschen klein, Sirene, Klavier, eigene Melodie	Einzel/Partner
5 min	Sicherung	Zwei, drei eigene Melodien vorspielen	Plenum

Typische Fehler

Beobachtung	Ursache	Hilfe
Man hört nur einen einzigen Ton.	Die „warte“-Blöcke fehlen; jeder Ton wird sofort vom nächsten abgelöst.	Nach jedem Ton ein „warte“ (Buch 9.3).
Die Sirene hört nicht auf.	„Ton stoppen“ fehlt nach der Schleife.	Block anhängen; im Notfall ■ **Stop** – das schaltet im Emulator auch den Ton ab.
Die Sirene klingt fast nur mit 600 Hz.	Das „warte“ nach dem zweiten Dauerton fehlt: 800 Hz wird sofort wieder von 600 Hz abgelöst.	Reihenfolge Dauerton – warte – Dauerton – warte.
Das Lied klingt „verwaschen“, Noten gleicher Höhe verschmelzen.	Ton genauso lang wie das „warte“.	Ton 50 ms kürzer als das Warten.
Kein Ton zu hören.	Lautstärke des Rechners/Browsers stumm oder Kopfhörer nicht eingesteckt.	Mit Aufgabe 1 prüfen.
Mini-Klavier: Der Ton hört nach dem Loslassen nicht auf.	Der innerste sonst-Teil mit „Ton stoppen“ fehlt.	„falls / sonst“ statt „falls“ verwenden.

Differenzierung

- **Schneller fertig:** Buch Kapitel 9, Aufgabe 6 (Tempo in einer Variable `takt`) oder das Mini-Klavier aus dem Buch mit sechs Tönen und RGB-LED (`k09_klavier.pbcp`).
- **Mehr Unterstützung:** Für Aufgabe 3 nur die ersten sechs Noten („Hänschen klein, ging allein“) bauen lassen.

LEHRERMATERIAL · AB 11

Turtle 1: Vielecke

Überblick

Lernziel	Die Schülerinnen und Schüler steuern eine Turtle mit „vorwärts“ und „drehen“, malen Vielecke mit einer Schleife und entdecken die Regel Drehwinkel = 360° ÷ Anzahl der Ecken.
Dauer	45 Minuten (Aufgabe 1–3 Pflicht, 4–5 Differenzierung)
Voraussetzungen	Blöcke holen und verbinden (AB 01), „wiederhole … -mal“ (AB 05)
Material	PBC+ Studio (Build 109 oder neuer), keine Bilder nötig. Lösungsdateien auf Anfrage
Buch	Kapitel 10.1, 10.2 und 10.4

Vorbereitung: Keine. Für den Einstieg eignet sich ein Stück Kreppband auf dem Boden oder ein freier Platz im Raum, auf dem ein Kind „Turtle“ spielt.

Stundenverlauf

Zeit	Phase	Inhalt	Sozialform
5 min	Einstieg	Ein Kind läuft als „Turtle“ nach Ansage ein Quadrat ab („4 Schritte vor, Vierteldrehung rechts …“). Frage: „Wie oft hat es sich gedreht? Wohin blickt es am Ende?“	Plenum
10 min	Aufgabe 1–2	Turtle erstellen (Name `kroete`), Quadrat nachbauen, mit Schleife verkürzen	Einzel/Partner
15 min	Aufgabe 3	Vielecke ausprobieren, Tabelle ausfüllen, Regel finden	Partner
10 min	Aufgabe 4/5, Nachgedacht	Stern, Muster mit verschachtelten Schleifen	Einzel/Partner
5 min	Sicherung	Tabelle vergleichen: Die Turtle dreht sich insgesamt einmal ganz herum (360°). Nachgedacht besprechen: Sie dreht sich um den **Außenwinkel**.	Plenum

Typische Fehler

Beobachtung	Ursache	Hilfe
Beim Klick auf einen Turtle-Befehl erscheint „Turtle fehlt“.	Es gibt noch keine Turtle.	Zuerst „neue Turtle erstellen“ anklicken und einen Namen eingeben.
Beim Testen erscheint „Turtle fehlt … Bevor du sie benutzt, musst du sie mit einem „neue Turtle“-Block erstellen.“	Der Block „neue Turtle“ wurde gelöscht oder hängt nicht unter „Beim Start“.	Block „neue Turtle“ direkt unter „Beim Start“ hängen.
Die Turtle heißt plötzlich `kr_te`.	Beim Namen wurde „kröte“ eingegeben; PBC+ Studio ersetzt Umlaute durch `_`.	Namen ohne Umlaute wählen: `kroete`.
Das Vieleck schließt sich nicht.	Falscher Drehwinkel (z. B. 60° beim Dreieck) oder falsche Anzahl Wiederholungen.	Tabelle aus Aufgabe 3 nutzen; Turtle-Spiel im Raum wiederholen.
Ein Teil der Figur fehlt.	Die Figur ragt über den Rand (die Turtle startet in der Mitte).	Kürzere Seiten wählen oder mit „Position auf x … y …“ woanders starten.
Beim Muster entsteht nur ein seltsamer Stern.	„rechts drehen um 30“ steckt in der inneren statt in der äußeren Schleife.	Block unter die innere Schleife ziehen, aber in der äußeren lassen.
Die Turtle malt sehr langsam (auf dem Gerät).	Geschwindigkeit 1 oder 2; auf dem PBC+ ist die Turtle dann deutlich langsamer als im Emulator.	„Geschwindigkeit 10“ oder 0 (sofort).

Differenzierung

- **Schneller fertig:** Aufgabe 5 mit eigenen Formen (Sechsecke, Sterne); Buch Kapitel 10, Aufgabe 3 (Kreis aus 36 Ecken) und Aufgabe 6 (Vieleck-Maschine).
- **Mehr Unterstützung:** `ab11_a2.pbcp` als Startpunkt geben; in Aufgabe 3 zunächst nur Dreieck und Sechseck ausprobieren lassen.

LEHRERMATERIAL · AB 12

Turtle 2: Spiralen und Muster

Überblick

Lernziel	Die Schülerinnen und Schüler verändern eine Variable in einer Schleife schrittweise und nutzen sie als Linienlänge. Sie setzen Stiftfarbe (auch zufällig) und Geschwindigkeit der Turtle ein.
Dauer	45 Minuten (Aufgabe 1–3 Pflicht, 4–6 Differenzierung)
Voraussetzungen	Turtle erstellen und steuern (AB 11), Variablen (AB 06), Zufallszahl (AB 07)
Material	PBC+ Studio (Build 109 oder neuer). Lösungsdateien auf Anfrage
Buch	Kapitel 10.3 und 10.4

Vorbereitung: Keine. Für den Einstieg die Lösung von Aufgabe 3 im Emulator zeigen (Beamer).

Stundenverlauf

Zeit	Phase	Inhalt	Sozialform
5 min	Einstieg	Bunte Spirale vorführen. „Die Schleife macht immer dasselbe – warum wird die Linie trotzdem länger?“	Plenum
10 min	Aufgabe 1	Variable `laenge` anlegen, Spirale nachbauen	Einzel
10 min	Aufgabe 2–3	Winkel verändern, Beobachtungen notieren; Zufallsfarbe	Partner
15 min	Aufgabe 4–6	Geschwindigkeit, runde Spirale, Blume mit „falls/sonst“	Einzel/Partner
5 min	Sicherung	Nachgedacht: Die Variable merkt sich die Länge; „setze laenge auf laenge + 5“ macht sie in jeder Runde größer.	Plenum

Typische Fehler

Beobachtung	Ursache	Hilfe
Alle Linien sind gleich lang (es entsteht ein Quadrat).	„setze laenge auf laenge + 5“ fehlt, darin steht eine feste Zahl statt `laenge`, oder „setze laenge auf 5“ steht **in** der Schleife.	Startwert vor die Schleife; in der Schleife den Rechenblock aus **Mathe** mit `laenge` links.

Beobachtung	Ursache	Hilfe
Die Spirale ist sofort riesig und läuft aus dem Bild.	Zu großer Zuwachs oder zu viele Wiederholungen.	Zuwachs oder Anzahl verkleinern.
Manche Linien sind kaum zu sehen.	Zufallsfarbe nahe Schwarz (alle drei Werte klein).	Blau fest auf 255 lassen oder die Zufallszahlen erst bei 50 beginnen.
Die Blume ist nur gelb.	Im „falls“ wird `i` mit 2 verglichen statt der Rest von `i` ÷ 2 mit 0.	Rechenblock „Rest“ verwenden: „i Rest 2 = 0“.

Differenzierung

- **Schneller fertig:** Eigene Muster: Spirale aus Sechsecken, Farbwechsel mit drei Farben (Rest von i ÷ 3); Buch Kapitel 10, Aufgabe 6 (Vieleck-Maschine).
- **Mehr Unterstützung:** `ab12_a1.pbcp` als Startpunkt geben; in Aufgabe 2 nur die Winkel 120 und 91 ausprobieren.

Turtle 3: Die Turtle sieht Farben

Überblick

Lernziel	Die Schülerinnen und Schüler fragen mit „Turtle … berührt Farbe …“ eine Farbe vor der Turtle ab und reagieren darauf in „solange“- und „falls“-Blöcken. Sie erklären, warum Hindernisse dicker als ein Schritt sein müssen.
Dauer	45 Minuten (Aufgabe 1–3 Pflicht, 4–6 Differenzierung)
Voraussetzungen	Turtle steuern (AB 11), falls und solange (AB 08/09), Zufallszahl (AB 07)
Material	PBC+ Studio (Build 109 oder neuer). Lösungsdateien auf Anfrage
Buch	Kapitel 10.5 (die Canvas-Rechtecke werden in Kapitel 11 erklärt)

Vorbereitung: Wer wenig Zeit hat, gibt `ab13_a1.pbcp` (Wände und Turtle) als Datei aus – das Abtippen der vier Rechtecke dauert sonst etwa 5 Minuten.

Stundenverlauf

Zeit	Phase	Inhalt	Sozialform
5 min	Einstieg	Staubsauger-Roboter (Lösung A4) im Emulator zeigen. „Woher weiß die Turtle, dass sie an der Wand ist?“	Plenum
10 min	Aufgabe 1	Wände auf den Canvas malen, Turtle erstellen	Einzel
10 min	Aufgabe 2–3	Bis zur Wand fahren, umdrehen, endlos hin und her	Einzel/Partner
15 min	Aufgabe 4–6	Zufallswinkel, dünne Wand, Zielfeld mit Zähler	Partner
5 min	Sicherung	Bild „3 Pixel vor der Turtle“ besprechen; Merksatz: Wände dicker als ein Schritt, Stiftfarbe ≠ Wandfarbe.	Plenum

Typische Fehler

Beobachtung	Ursache	Hilfe
Die Turtle fährt durch die Wand.	Die Wand hat eine andere Farbe als im Block abgefragt (z. B. Rot aus dem Farbwähler statt „Farbe rot“).	Beide Male „Farbe nach Namen: rot“ verwenden.

Beobachtung	Ursache	Hilfe
Die Turtle bleibt an der ersten Wand hängen und dreht sich auf der Stelle.	Stiftfarbe rot: Nach dem Umdrehen sieht die Turtle ihre eigene rote Linie.	Andere Stiftfarbe (gelb).
Man sieht die Wände nicht, aber die Turtle dreht trotzdem um.	Die Rechtecke stehen **nach** „neue Turtle“. Sie sind auf dem Canvas, werden im Emulator aber erst mit „Canvas zeigen“ sichtbar.	Wände vor „neue Turtle“ malen.
Vor dem Start fragt PBC+ Studio „Endlosschleife ohne Pause“.	In „unendlich wiederholen“ fehlt ein „warte“-Block.	„warte 10 ms“ in die Schleife.
Die Turtle fährt einmal los und reagiert nie auf die Wand.	Der „falls“-Block steht vor oder nach der Schleife statt darin.	„falls“ in die Schleife ziehen.
Der Text am Ziel erscheint nicht.	„Canvas zeigen“ fehlt nach „Text auf Canvas“.	Block anhängen.
Im Zieltext erscheinen Fragezeichen.	Umlaute im Text (z. B. „Stöße“).	„ue“, „oe“ schreiben oder andere Wörter wählen.

Differenzierung

- **Schneller fertig:** Zwei Turtles mit verschiedenen Farben fahren im selben Raum (zweite Turtle mit „neue Turtle erstellen“); Hindernisse in der Mitte des Raums.
- **Mehr Unterstützung:** `ab13_a1.pbcp` als Startpunkt; Aufgabe 2 gemeinsam am Beamer bauen.

Mein Hintergrund

Überblick

Lernziel	Die Schülerinnen und Schüler malen mit Canvas-Blöcken ein Bild im Speicher und zeigen es mit „Canvas zeigen“. Sie laden ein Bild hoch und legen es mit „Bild auf Canvas“ als Hintergrund an.
Dauer	45 Minuten (Aufgabe 1–3 Pflicht, 4–6 Differenzierung)
Voraussetzungen	Zeichnen und Farben (AB 03), Koordinaten (AB 02), Zufallszahl (AB 07) für Aufgabe 6
Material	PBC+ Studio (Build 109 oder neuer), Bild `weltraum.png` aus dem Material-Paket. Lösungsdateien auf Anfrage
Buch	Kapitel 11

Vorbereitung: `weltraum.png` auf die Schülerrechner kopieren oder im Netzwerk bereitstellen. Im Emulator reicht das Hochladen im Browser (**Dateien → Bilder verwalten…**); wer mit Geräten arbeitet, lädt das Bild auch auf jedes Gerät. AB 15 braucht dasselbe Bild.

Stundenverlauf

Zeit	Phase	Inhalt	Sozialform
5 min	Einstieg	Bild „Maler und Leinwand“: Erst wird gemalt, dann wird das fertige Bild aufgehängt. Abbildung aus Buch 11.1 zeigen.	Plenum
10 min	Aufgabe 1–2	Landschaft nachbauen; ohne „Canvas zeigen“ testen	Einzel
15 min	Aufgabe 3	Eigenes Bild ergänzen	Einzel
10 min	Aufgabe 4–6	Hintergrundbild, eigenes Bild stempeln, Sternenhimmel	Einzel/Partner
5 min	Sicherung	Nachgedacht: Canvas = Bild im Speicher, „Canvas zeigen“ bringt es auf einmal auf den Bildschirm. Ausblick: Sprites vor dem Hintergrund (AB 15).	Plenum

Typische Fehler

Beobachtung	Ursache	Hilfe
Der Bildschirm bleibt schwarz.	„Canvas zeigen“ fehlt oder hängt nicht unten am Programm.	Block ans Ende hängen.
Ein Text erscheint kurz und ist dann weg, oder er ist nicht zu sehen.	Es wurde der Text-Block aus **Display** genommen. „Canvas zeigen“ malt danach das ganze Bild darüber.	Text-Block aus der Kategorie **Canvas** nehmen.
Haus oder Sonne sind nicht zu sehen.	Sie wurden **vor** dem Himmel gemalt; „Canvas füllen“ übermalt alles.	„Canvas füllen“ an den Anfang.
Im Text erscheinen Fragezeichen.	Umlaute oder ß.	„ue“, „oe“, „ss“ schreiben.
Das Bild fehlt in der Liste des Blocks „Bild auf Canvas“.	Bild noch nicht hochgeladen.	**Dateien → Bilder verwalten…**
Beim Testen erscheint ein `NameError`.	Der Block „Canvas anlegen“ steht unter anderen Canvas-Blöcken.	„Canvas anlegen“ löschen – der Canvas wird automatisch angelegt.

Differenzierung

- **Schneller fertig:** Buch Kapitel 11, „Ausprobieren: Der Wetterbericht“ und Aufgabe 1 (Flagge).
- **Mehr Unterstützung:** `ab14_a1.pbcp` als Startpunkt geben und in Aufgabe 3 nur die Zahlen im Rechteck-Block ändern lassen.

LEHRERMATERIAL · AB 15

Sprites in Bewegung

Überblick

Lernziel	Die Schülerinnen und Schüler legen Sprites an, setzen, zeigen und bewegen sie (gleiten, versetzen), steuern ein Sprite mit dem Joystick und fragen seine Position mit „x von Sprite“ ab. Sie nutzen ein Bild auf dem Canvas als Hintergrund.
Dauer	45 Minuten (Aufgabe 1–3 Pflicht, 4–5 Differenzierung)
Voraussetzungen	Canvas und „Bild auf Canvas“ (AB 14), „wird … gedrückt?“ (AB 09), Variablen und falls (AB 06/08)
Material	PBC+ Studio (Build 109 oder neuer), Bilder `rakete.png`, `ufo.png`, `weltraum.png` aus dem Material-Paket. Lösungsdateien auf Anfrage
Buch	Kapitel 12

Vorbereitung: Die drei Bilder auf die Schülerrechner kopieren oder im Netzwerk bereitstellen. Beim Öffnen einer Lösungsdatei sind die Sprites schon angelegt; die Bilder müssen trotzdem vorher hochgeladen sein.

Stundenverlauf

Zeit	Phase	Inhalt	Sozialform
5 min	Einstieg	Lösung A4 vorführen. „Was ist anders als bei den Bildern auf dem Canvas?“ – Sprites bewegen sich, der Hintergrund bleibt heil.	Plenum
8 min	Vorbereitung	Bilder hochladen, Sprites `rakete` und `ufo` anlegen	Einzel
7 min	Aufgabe 1	Rakete setzen, zeigen, gleiten lassen	Einzel
10 min	Aufgabe 2–3	Joystick-Steuerung, Hintergrundbild	Einzel/Partner
10 min	Aufgabe 4–5	Ufo mit `schritt` und „x von Sprite“, Randbegrenzung	Partner
5 min	Sicherung	Nachgedacht: „gleite“ hält das Programm an. Merksatz: Für Steuerung in der Schleife „versetze“ nehmen.	Plenum

Typische Fehler

Beobachtung	Ursache	Hilfe
Meldung „Sprite fehlt“ beim Holen eines Sprite-Blocks	Noch kein Sprite angelegt.	Zuerst „+ neues Sprite…“.
Im Dialog steht „— Noch keine Bilder hochgeladen —“.	Bilder nicht hochgeladen.	**Dateien → Bilder verwalten…**, danach Dialog neu öffnen.
„Ungültiger Name …“ beim Anlegen	Umlaute, Leerzeichen oder Zahl am Anfang (z. B. „Räkete“, „1ufo“).	Nur Buchstaben, Ziffern, `_`; nicht mit Ziffer beginnen.
Die Rakete blitzt kurz oben links auf.	„zeige Sprite“ steht vor „setze Sprite“; ein neues Sprite steht bei (0 | 0).	Erst setzen, dann zeigen.
Die Rakete fliegt in die falsche Richtung.	Vorzeichen vertauscht: Nach oben heißt y **kleiner**.	Tabelle an die Tafel: hoch (0, −3), runter (0, 3).
Ein Text oder eine Form verschwindet, sobald sich die Rakete bewegt.	Mit **Display**-Blöcken gemalt; bei jeder Sprite-Bewegung wird der Bildschirm neu gemalt.	Auf den **Canvas** malen bzw. schreiben.
Das Ufo fliegt aus dem Bild und kommt nie zurück.	Im zweiten „falls“ wird `schritt` auf −2 statt 2 gesetzt (oder `schritt` nie auf 2 gesetzt).	Werte prüfen: rechts → −2, links → 2.
Vor dem Start fragt PBC+ Studio „Endlosschleife ohne Pause“.	„warte 20 ms“ fehlt in der Schleife.	Block in die Schleife ziehen.

Differenzierung

- **Schneller fertig:** Buch Kapitel 12, Aufgabe 6 (Das Ufo verfolgt die Rakete); danach weiter mit AB 16 (Kollisionen).
- **Mehr Unterstützung:** `ab15_a2.pbcp` als Startpunkt geben; Aufgabe 4 mit der fertigen Datei `ab15_a4.pbcp` untersuchen statt selbst bauen.

Kollision 1: Achtung, Hai!

Überblick

Lernziel	Die Schülerinnen und Schüler erkennen mit dem Block „Sprite … berührt Sprite …“, ob sich zwei Sprites berühren, und reagieren darauf in einer Spielschleife. Sie beenden ein Spiel mit einer Variable und „solange“.
Dauer	45 Minuten (Aufgabe 1–3 Pflicht, 4–5 Differenzierung)
Voraussetzungen	Sprites anlegen und bewegen, „x von Sprite“ (AB 15), Variablen (AB 06/07), falls und solange (AB 08/09), Canvas füllen (AB 14)
Material	PBC+ Studio (Build 109 oder neuer), Bilder `fisch.png` und `hai.png` aus dem Material-Paket, Lösungsdateien auf Anfrage
Buch	Kapitel 13.1 und 13.4

Vorbereitung durch die Lehrkraft: Die beiden Bilder auf die Schülerrechner kopieren oder im Netzwerk bereitstellen. `fisch.png` ist derselbe Fisch wie in Kapitel 12 / AB 15. Wer mit Geräten arbeitet: Die Bilder über **Dateien → Bilder verwalten…** auf jedes Gerät laden. Im Emulator reicht das Hochladen im Browser.

> Im Emulator läuft das Spiel langsamer als auf dem Gerät, weil ein mit Blöcken bemalter Canvas (hier das blaue Meer) im Browser mehr Rechenzeit braucht. Das ist normal und kein Fehler im Programm.

Stundenverlauf

Zeit	Phase	Inhalt	Sozialform
5 min	Einstieg	Fertiges Spiel vorführen (Lösung A4 im Emulator). Frage: „Woher weiß das Programm, dass der Hai den Fisch erwischt hat?“	Plenum
5 min	Vorbereitung	Bilder hochladen, Sprites `fisch` und `hai` anlegen	Einzelarbeit
10 min	Aufgabe 1–2	Fisch steuern, Hai schwimmen lassen	Einzel/Partner
10 min	Aufgabe 3	Kollision einbauen und testen	Einzel/Partner

Zeit	Phase	Inhalt	Sozialform
10 min	Aufgabe 4/5	Leben zählen, Spielende; schnelle SuS: Tempo steigern	Einzel/Partner
5 min	Sicherung	„Nachgedacht“ besprechen: Sprites sind Rechtecke. Merksatz: Die Frage muss in der Schleife stehen.	Plenum

Typische Fehler

Beobachtung	Ursache	Hilfe
Der Hai erwischt den Fisch nie.	Der Kollisions-„falls“ steht **vor** oder **unter** der Schleife und wird nur einmal geprüft.	Block in die Schleife ziehen, direkt über „warte“.
Der Hai taucht nie auf, der Ton piept ständig.	Im Block „berührt Sprite“ ist zweimal `hai` (oder zweimal `fisch`) gewählt – ein Sprite berührt sich immer selbst.	Dropdowns prüfen: `fisch` berührt `hai`.
Der Hai taucht nie auf, ohne Ton.	Im Vergleich steht „>“ statt „<“: x von hai > −40 ist sofort wahr, der Hai wird in jedem Durchlauf wieder nach rechts gesetzt.	Vergleich auf „<“ umstellen.
Der Hai schwimmt nach rechts weg und kommt nie wieder.	„versetze hai um 4“ statt um −4.	Minuszeichen ergänzen.
„Game Over“ erscheint sofort beim Start.	`leben` wird nicht auf 3 gesetzt (startet bei 0), die Schleife läuft gar nicht.	„setze leben auf 3“ vor die Schleife.
Das Spiel endet nie, obwohl der Hai trifft.	„setze leben auf 3“ steht **in** der Schleife, oder „unendlich wiederholen“ wurde nicht ersetzt.	Startwert vor die Schleife; „solange leben > 0“ verwenden.
Ein Sprite ist unsichtbar.	„zeige Sprite“ fehlt.	Am Anfang beide Sprites zeigen.

Differenzierung

- **Schneller fertig:** Aufgabe 5; danach Buch Kapitel 13, Aufgabe 5 („Ausgewichen!“: Punkte für jeden Hai, der vorbeischwimmt).
- **Mehr Unterstützung:** Die Datei `ab16_a1.pbcp` als Startpunkt geben, dann direkt mit Aufgabe 2 beginnen.

LEHRERMATERIAL · AB 17

Kollision 2: Vorsicht, Lava!

Überblick

Lernziel	Die Schülerinnen und Schüler malen Hindernisse auf den Hintergrund und erkennen sie mit „Sprite … berührt Farbe …“. Sie halten ein Sprite mit „Sprite … berührt Rand“ auf dem Bildschirm und beenden ein Spiel mit „solange nicht …“.
Dauer	45 Minuten (Aufgabe 1–3 Pflicht, 4–5 Differenzierung)
Voraussetzungen	AB 16 (Kollision in der Spielschleife), Canvas-Rechteck (AB 14), Steuern mit dem Joystick (AB 09/15)
Material	PBC+ Studio (Build 109 oder neuer), Bild `held.png` aus dem Material-Paket, Lösungsdateien auf Anfrage
Buch	Kapitel 13.2 bis 13.4

Vorbereitung durch die Lehrkraft: `held.png` bereitstellen. Für Gruppen, die mehr Unterstützung brauchen, `ab17_a1.pbcp` (die Vorlage aus Aufgabe 1) bereithalten – das Nachbauen der Steuerung kostet sonst 10 Minuten.

> „berührt Farbe“ vergleicht die Farbe **exakt**. Die Lava wird mit „Farbe rot“ gemalt und mit „Farbe rot“ geprüft. Ein Rot aus dem Farbwähler wird nur erkannt, wenn es genau dieselbe Farbe ist. Geprüft wird nur der Hintergrund (Canvas), nicht andere Sprites. Im Emulator läuft das Programm langsamer als auf dem Gerät, weil der bemalte Canvas im Browser mehr Rechenzeit braucht.

Stundenverlauf

Zeit	Phase	Inhalt	Sozialform
5 min	Einstieg	Bild der Lava-Strecke zeigen. „Wie kann das Programm merken, dass der Held in die Lava tritt? Er berührt ja kein Sprite.“	Plenum
15 min	Aufgabe 1	Bild hochladen, Sprite anlegen, Vorlage nachbauen	Einzel/Partner
10 min	Aufgabe 2–3	Lava und Rand als Kollision; Experiment mit falscher Farbe	Einzel/Partner
10 min	Aufgabe 4/5	Zielfeld und Spielende; schnelle SuS: Superkraft	Einzel/Partner

Zeit	Phase	Inhalt	Sozialform
5 min	Sicherung	Drei Kollisions-Blöcke vergleichen (Sprite, Rand, Farbe). Aufgabe 5 besprechen: Die Frage ist eine Momentaufnahme.	Plenum

Typische Fehler

Beobachtung	Ursache	Hilfe
Die Lava wird nicht erkannt.	Im Kollisions-Block ist eine andere Farbe eingestellt als beim Malen (z. B. Farbwähler statt „Farbe rot“).	Beide Male „Farbe nach Namen: rot“ verwenden.
Die Lava wird nicht erkannt, obwohl die Farbe stimmt.	Der „falls“-Block steht vor oder unter der Schleife.	In die Schleife ziehen, direkt über „warte“.
Der Held springt dauernd zum Start (bei Lava mit Ton).	Die Startposition liegt in der Lava oder ragt über den Rand.	Startposition prüfen, z. B. x 20, y 20.
„Geschafft!“ erscheint sofort beim Start.	Das „nicht“ fehlt: „solange held berührt grün“ ist am Anfang falsch, die Schleife läuft gar nicht.	„nicht“ aus der Kategorie Mathe einsetzen.
Das Zielfeld fehlt.	Der Rechteck-Block hängt nicht in der Kette unter „Beim Start“.	Block an die Lava-Blöcke anhängen.
Der Held rast bei Taste A nach unten aus dem Bild.	„wird A gedrückt?“ statt „wurde A seit der letzten Abfrage gedrückt?“ – der Sprung wiederholt sich, solange die Taste unten ist.	Den Block „wurde … gedrückt?“ nehmen.

Differenzierung

- **Schneller fertig:** Ein eigenes Lava-Labyrinth mit mehr Rechtecken malen und von der Nachbarin oder dem Nachbarn spielen lassen. Oder: Buch Kapitel 13, Aufgabe 4 (Ziellinie mit Zeitmessung auf der Rennstrecke).
- **Mehr Unterstützung:** Die Datei `ab17_a1.pbcp` als Startpunkt geben und direkt mit Aufgabe 2 beginnen.

LEHRERMATERIAL · AB 18

Ordnung mit Funktionen

Überblick

Lernziel	Die Schülerinnen und Schüler teilen ein vorhandenes Programm in benannte Funktionen auf, rufen sie auf und geben einer Funktion einen Parameter.
Dauer	45 Minuten (Aufgabe 1–3 Pflicht, 4–5 Differenzierung)
Voraussetzungen	Münzspiel aus AB 16 (Sprites, „berührt Sprite“, Variable `punkte`, „solange“)
Material	PBC+ Studio (Build 109 oder neuer), Bilder `held.png`, `muenze.png` (für Aufgabe 5 auch `geist.png`) aus dem Material-Paket, Vorlage `ab18_vorlage.pbcp`, Lösungsdateien auf Anfrage
Buch	Kapitel 14.1, 14.2 und 14.4

Vorbereitung durch die Lehrkraft: Die Vorlage `ab18_vorlage.pbcp` und die drei Bilder auf die Schülerrechner kopieren. Die Vorlage enthält schon die Sprites `held` und `muenze`; die Bilder müssen aber vor dem Testen im Browser hochgeladen sein (**Dateien → Bilder verwalten…**).

Stundenverlauf

Zeit	Phase	Inhalt	Sozialform
5 min	Einstieg	Die Vorlage am Beamer zeigen: „Findet ihr auf einen Blick, wo die Münze geprüft wird?“ Vergleich mit einem Buch ohne Kapitel.	Plenum
8 min	Aufgabe 1	Vorlage öffnen, testen, die vier Teile benennen	Partnerarbeit
12 min	Aufgabe 2	Erste Funktion `steuern` anlegen, Blöcke verschieben, Aufruf einsetzen. Lehrkraft macht Schritt 2 und 3 einmal vor (Blöcke hängen zusammen!).	Einzelarbeit
12 min	Aufgabe 3–5	Restliche Funktionen; schnelle SuS: Parameter, Geist	Einzel/Partner
8 min	Sicherung	Ein aufgeräumtes „Beim Start“ zeigen. Frage: „Warum sind Funktionen praktisch?“ (lesbar, wiederverwendbar, Fehler leichter finden). Merksatz: Eine Funktion tut nur etwas, wenn sie aufgerufen wird.	Plenum

Typische Fehler

Beobachtung	Ursache	Hilfe
Nach dem Verschieben bewegt sich der Held nicht mehr.	Funktion `steuern` angelegt, aber der Aufruf fehlt in der Schleife.	Aufruf aus der Kategorie **Funktionen** holen und in die Schleife setzen.
Die Münze wird nicht mehr geprüft oder „warte“ fehlt in der Schleife.	Beim Ziehen kamen alle Blöcke darunter mit in die Funktion.	Den „falls … berührt“-Block (mit „warte“) zurück in die Schleife ziehen.
Beim Testen erscheint „Funktionsaufruf ohne Funktion“.	Die Funktion wurde umbenannt oder gelöscht. Aufrufe benennen sich **nicht** mit um und zeigen dann „name (?)“.	Im Aufruf über das Feld ▾ den neuen Namen wählen.
Aus `münze_prüfen` wird `m_nze_pr_fen`.	PBC+ Studio ersetzt Umlaute und Leerzeichen in Funktionsnamen durch `_`.	Gleich `muenze_pruefen` schreiben (ins Namensfeld des Funktions-Blocks klicken).
Nach dem Einbau von `piep` ist beim Münzfang kein Piep zu hören.	Die Lücke `hoehe:` im Aufruf ist leer – der Parameter hat dann keinen Wert.	Einen Block „Zahl“ aus **Mathe** in die Lücke ziehen.
Meldung „Unbekannter Name“ mit `hoehe`.	Der Parameter-Block `hoehe` wurde außerhalb der Funktion `piep` benutzt.	Parameter-Blöcke nur innerhalb ihrer Funktion benutzen.
Meldung „Endlosschleife ohne Pause“, obwohl in einer Funktion ein „warte“ steht.	Wer „unendlich wiederholen“ benutzt: Die IDE sucht das „warte“ nur direkt in der Schleife, nicht in aufgerufenen Funktionen.	„warte 20 ms“ in die Schleife selbst setzen (oder mit OK bestätigen).

Differenzierung

- **Schneller fertig:** Aufgabe 5; danach Buch Kapitel 14, Aufgaben 3 und 6.
- **Mehr Unterstützung:** Mit `ab18_a2.pbcp` weitermachen – dort ist `steuern` schon fertig, die SuS bauen nur die übrigen Funktionen nach diesem Muster.

Listen 1: Meine Bestenliste

Überblick

Lernziel	Die Schülerinnen und Schüler legen eine Liste an, hängen Werte an, lesen ihre Länge, zeigen alle Elemente mit „für jedes Element“ an und bestimmen den größten Wert. Sie wissen, dass Listen ab 0 nummeriert sind.
Dauer	45 Minuten (Aufgabe 1–4 Pflicht, 5–6 Differenzierung)
Voraussetzungen	Variablen, „falls“, Funktionen aufrufen (AB 18)
Material	PBC+ Studio (Build 109 oder neuer), Vorlage `ab19_vorlage.pbcp`, Lösungsdateien auf Anfrage
Buch	Kapitel 15.1 bis 15.4

Vorbereitung durch die Lehrkraft: Die Vorlage auf die Schülerrechner kopieren. Bilder werden nicht gebraucht. Achtung: Die Dateinamen der Lösungen sind gegen die Aufgabennummern um eins verschoben (die Vorlage ist Stufe 1).

Stundenverlauf

Zeit	Phase	Inhalt	Sozialform
5 min	Einstieg	Klick-Rennen am Beamer gegen eine Schülerin/einen Schüler. „Wie merken wir uns die Ergebnisse von 30 Runden? 30 Variablen?“	Plenum
5 min	Aufgabe 1	Vorlage testen, Problem formulieren	Partnerarbeit
5 min	Aufgabe 2	Fächer-Modell auf Papier; kurz im Plenum vergleichen: Zählung ab 0	Einzelarbeit
20 min	Aufgabe 3–6	Liste anlegen, anhängen, anzeigen, Rekord	Einzel/Partner
10 min	Sicherung	Bestenlisten vergleichen. Merksätze: Erstes Element = Nr. 0, letztes = Länge − 1; „für jedes Element“ geht alle Werte durch.	Plenum

Typische Fehler

Beobachtung	Ursache	Hilfe
Meldung „Liste fehlt“ beim Holen eines Listen-Blocks	Noch keine Liste angelegt.	Erst „+ neue Liste…“ in der Kategorie **Listen**.
Es steht immer nur ein Ergebnis da, „Runde“ bleibt 1.	„setze Liste ergebnisse auf leer“ steht **in** der Schleife.	Den Block vor „unendlich wiederholen“ ziehen.
Das erste Ergebnis ist 0, alle anderen sind um eine Runde verschoben.	„hänge … an“ steht **über** dem Aufruf `runde_spielen`.	Unter den Aufruf ziehen.
Der Rekord ist immer das letzte Ergebnis.	„falls element > rekord“ steht unter statt in der Schleife, oder „setze rekord auf 0“ steht in der Schleife.	Reihenfolge mit dem Bild vergleichen.
Die Ergebnisse stehen übereinander.	„setze y auf y + 20“ fehlt oder steht außerhalb der Schleife.	In die Schleife setzen.
Ein Klick wird zu viel gezählt.	„wurde A … gedrückt?“ merkt sich auch einen Druck auf A, der **vor** dem Start passiert ist.	Erklären; vor dem Start nicht auf A drücken.

Differenzierung

- **Schneller fertig:** Aufgabe 6; danach den Durchschnitt aller Ergebnisse anzeigen (Buch Kapitel 15, Aufgabe 4).
- **Mehr Unterstützung:** `ab19_a2.pbcp` (Aufgabe 3 fertig) als Start geben; Aufgabe 2 gemeinsam an der Tafel mit Kästchen lösen.

Listen 2: Merk dir die Folge

Überblick

Lernziel	Die Schülerinnen und Schüler speichern zusammengehörige Daten in mehreren Listen mit gleicher Nummer, greifen mit „Element Nr.“ darauf zu, lassen eine Liste wachsen und gehen sie mit „für jedes Element“ durch. Sie bauen eine Funktion mit Rückgabewert und lokaler Variable.
Dauer	90 Minuten (2 Stunden: Seite 1–2 und Seite 3–4; Aufgabe 1–5 Pflicht, 6–7 Differenzierung)
Voraussetzungen	Funktionen mit Parametern (AB 18), Listen anlegen und „für jedes Element“ (AB 19)
Material	PBC+ Studio (Build 109 oder neuer). Lösungsdateien auf Anfrage: `ab20_a1.pbcp` (Spielfeld) bis `ab20_a7.pbcp` (schneller werdend), je eine Datei pro Aufgabe
Buch	Kapitel 15 (besonders 15.2 und 15.6), Kapitel 14.3 und 14.4

Vorbereitung durch die Lehrkraft: Keine Bilder nötig. Für die zweite Stunde die Datei `ab20_a3.pbcp` bereithalten – wer in der ersten Stunde nicht fertig wurde, startet damit.

Stundenverlauf

Stunde 1

Zeit	Phase	Inhalt	Sozialform
5 min	Einstieg	Lösung `ab20_a6.pbcp` vorspielen, eine Schülerin spielt. „Was muss sich der PicoBoy merken?“ → die Folge, und sie wird immer länger: eine Liste.	Plenum
15 min	Aufgabe 1	Vier Listen, `zeichne_feld`, `spielfeld`. Klären: Fach Nr. 2 in allen vier Listen gehört zu **einem** Feld.	Einzelarbeit
10 min	Aufgabe 2	`aufleuchten` bauen und testen	Einzel/Partner
10 min	Aufgabe 3	Folge als Liste, `folge_zeigen`	Einzel/Partner
5 min	Sicherung	Folge [1, 1, 3, 0] gemeinsam „vorspielen“ (Schüler zeigen mit den Armen die Richtung)	Plenum

Stunde 2

Zeit	Phase	Inhalt	Sozialform
5 min	Wiederholung	Was steht schon? Was fehlt? (Nachspielen und Vergleichen)	Plenum
12 min	Aufgabe 4	`taste_lesen` mit Rückgabe und lokaler Variable	Einzelarbeit
15 min	Aufgabe 5	`nachspielen`, Spielschleife	Einzel/Partner
8 min	Aufgabe 6/7	Spielende, Punkte; schnelle SuS: Tempo, eigene Ideen	Einzel/Partner
5 min	Sicherung	Rekorde vergleichen. Welche Listen hat das Spiel, und wofür? (fx, fy, farben, toene = Eigenschaften der Felder; folge = Abfolge)	Plenum

Typische Fehler

Beobachtung	Ursache	Hilfe
Ein Listen-Block arbeitet mit der falschen Liste.	Neue Listen-Blöcke kommen immer mit der ersten Liste.	Im Block auf den Listennamen (▾) klicken und die richtige Liste wählen.
Meldung „Listen-Position außerhalb“	Ein „erzeuge Liste mit“ hat weniger als 4 Lücken, oder in `zeichne_feld` steht eine Nummer über 3.	Mit dem Zahnrad 4 Lücken einstellen, Werte mit dem Bild vergleichen.
Das Feld blitzt nicht weiß auf.	„warte“ nach dem Ton fehlt (Aufgabe 2).	„warte 300 ms“ einfügen.
Die Folge ist in jeder Runde nur 1 Element lang.	„setze Liste folge auf leer“ steht **in** der Schleife.	Vor die „solange“-Schleife ziehen.
Ein Druck zählt doppelt, oder das Spiel ist sofort verloren.	In `taste_lesen` wurde „wird … gedrückt?“ statt „wurde … gedrückt?“ benutzt. Ein gehaltener Joystick wird dann immer wieder gelesen.	Block „wurde … seit der letzten Abfrage gedrückt?“ nehmen.
`taste_lesen` endet nie oder gibt immer −1 zurück.	Die vier Blöcke „lokale Variable taste =“ setzen eine andere Variable (Tippfehler im Namen) oder fehlen in den „falls“-Blöcken.	Namen vergleichen; Blöcke mit Rechtsklick → Kopieren vervielfältigen.
Meldung „Endlosschleife ohne Pause“ beim Test in	In der „unendlich wiederholen“-Schleife steht kein „warte“. Die IDE sieht das „warte“	„warte 20 ms“ in die Schleife setzen (oder OK

Beobachtung	Ursache	Hilfe
Aufgabe 4	in den aufgerufenen Funktionen nicht.	klicken).
Der Aufruf „aufleuchten nr:“ hat eine leere Lücke.	Der Aufruf `taste_lesen` wurde daneben statt in die Lücke gelegt.	Den runden Block `taste_lesen` in die Lücke `nr:` ziehen.

Differenzierung

- **Schneller fertig:** Aufgabe 7 und die Ideen darunter, z. B. ein fünftes Feld (Taste A, Mitte) – dafür müssen alle vier Listen ein fünftes Element bekommen, `spielfeld` zählt bis 4, die Zufallszahl geht bis 4 und `taste_lesen` bekommt eine fünfte Abfrage.
- **Mehr Unterstützung:** In Stunde 2 mit `ab20_a3.pbcp` starten. Aufgabe 4 gemeinsam an der Tafel/am Beamer bauen, da hier Funktion mit Rückgabe und lokale Variable zusammenkommen.

LEHRERMATERIAL · AB 21

Kipp mich!

Überblick

Lernziel	Die Schülerinnen und Schüler lesen den Beschleunigungssensor mit dem Block „Beschleunigung … in …“ aus, deuten die Werte (Vorzeichen, Einheit) und steuern ein Sprite durch Kippen des Geräts.
Dauer	45 Minuten (Aufgabe 1–2 Pflicht, 3–5 Differenzierung)
Voraussetzungen	Sprites (AB 15), Kollisionen mit Rand und Farbe (AB 16/17), Variablen (AB 06)
Material	**PBC±Geräte mit USB-Kabel** (am besten eins pro Paar), PBC+ Studio in Chrome oder Edge, Bild `kugel.png` aus dem Material-Paket, Lösungsdateien auf Anfrage
Buch	Kapitel 18 – Der Beschleunigungssensor

Wichtig: Der Emulator hat keinen Sensor. Er liefert immer X = 0, Y = 0, Z = 1 g (bzw. 9,81 m/s²). Die Programme laufen dort ohne Fehler, aber die Kugel bewegt sich nicht. Dieses Arbeitsblatt ist nur **mit Geräten** sinnvoll. Eine Alternative ohne Geräte steht am Ende.

Vorbereitung durch die Lehrkraft: Geräte laden und mit aktueller Firmware versehen, Verbindung einmal testen. `kugel.png` bereitstellen. Wer will, lädt das Bild schon vorher auf jedes Gerät (**Dateien → Bilder verwalten…** bei verbundenem Gerät).

Stundenverlauf

Zeit	Phase	Inhalt	Sozialform
5 min	Einstieg	Ein Gerät mit `ab21_a4.pbcp` herumgeben: „Wie merkt das Gerät, dass ich es kippe?“ Kurz: Sensor spürt die Erdanziehung, drei Achsen (Buch, Abb. in 18).	Plenum
5 min	Vorbereitung	Gerät verbinden, Ziel PBC+, Bild hochladen, Sprite `kugel` anlegen	Partnerarbeit
10 min	Aufgabe 1	Messprogramm bauen, Tabelle ausfüllen	Partnerarbeit
10 min	Aufgabe 2	Kugel rollen lassen, Frage nach dem Minus	Partnerarbeit
10 min	Aufgabe 3–5	Rand, Loch, Spielende	Partnerarbeit

Zeit	Phase	Inhalt	Sozialform
5 min	Sicherung	Tabelle vergleichen: „Die Achse, die nach oben zeigt, ist positiv.“ Warum −5? Warum geht es im Emulator nicht?	Plenum

Typische Fehler

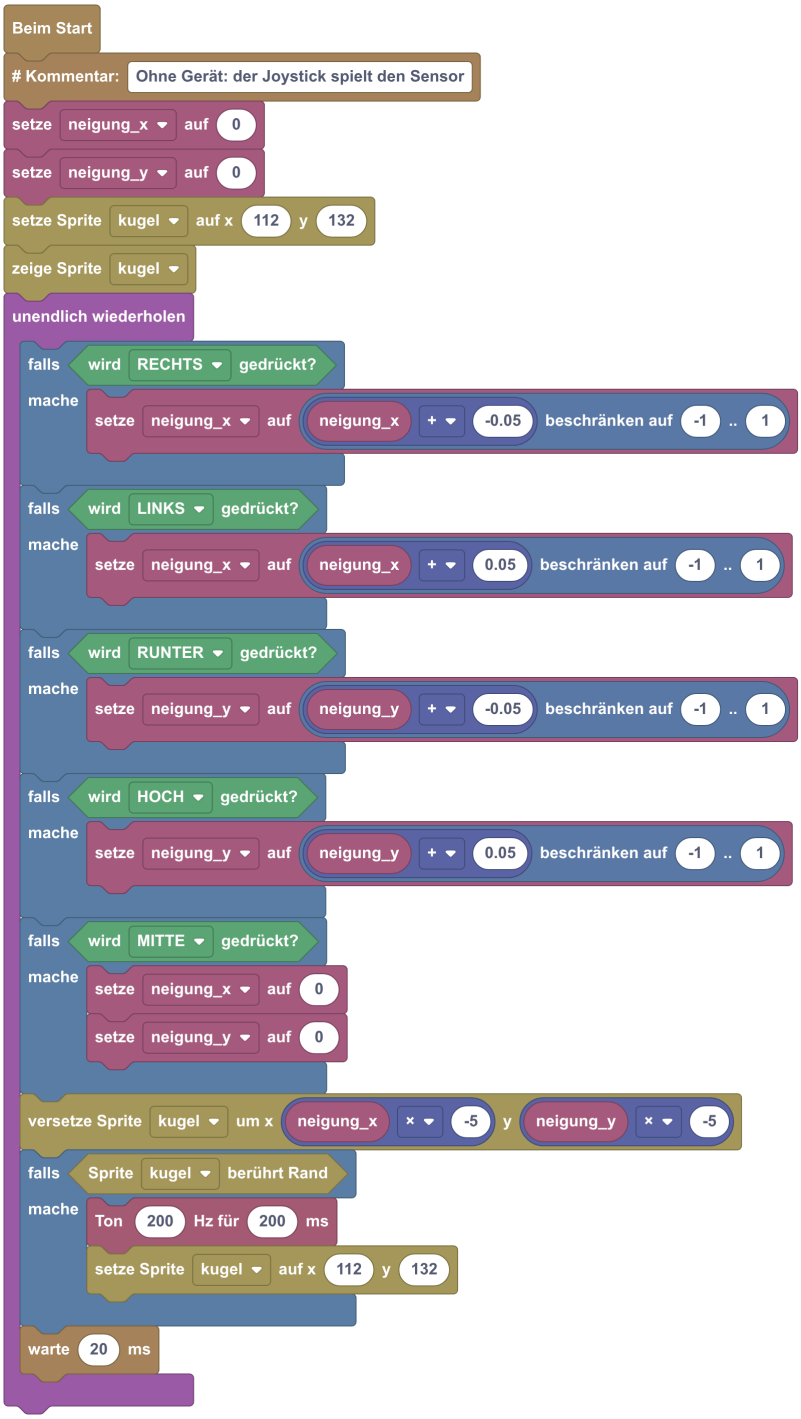
Beobachtung	Ursache	Hilfe
Die Kugel bewegt sich gar nicht, Werte immer 0 / 0.	Ziel steht auf **Emulator**.	Ziel **PBC+** wählen, verbinden, neu testen.
In Aufgabe 1 zeigt X fast immer 0, nur ganz schief −1 oder 1.	Einheit **g** statt **m/s²** gewählt – „runden“ macht aus −0,4 g eine 0.	Einheit auf m/s² stellen.
Die Kugel läuft immer schräg / nur in eine Richtung.	Im zweiten Beschleunigungs-Block steht noch X statt Y.	Achse im y-Feld auf **Y** stellen.
Die Kugel rollt bergauf.	Faktor +5 statt −5.	Minus ergänzen (Aufgabe 2 besprechen).
Bei kleiner Neigung passiert nichts.	Ganze Pixel (siehe Nachgedacht).	Kein Fehler. Wer will: Faktor −10.
Das Loch wird nie erkannt.	Kreis mit einem Grün aus dem Farbwähler gemalt, aber auf „Farbe grün“ geprüft (oder umgekehrt).	Beide Male „Farbe nach Namen: grün“.
Meldung „Endlosschleife ohne Pause“.	„warte 20 ms“ fehlt in der Schleife.	Block ergänzen.
Das USB-Kabel stört beim Kippen.	Programm läuft per ▶ **Testen** über das Kabel.	Programm mit **Dateien → Auf den PBC+…** als **Python-Programm** speichern und im Startmenü des Geräts starten (Buch, Kapitel 1.6).

Differenzierung

- **Schneller fertig:** Aufgabe 5 mit roter Lava vor dem Loch (Buch Kapitel 18: Kugel-Labyrinth, `k18_labyrinth.pbcp`) oder Buch Kapitel 18, Aufgabe 5 (Schüttel-Würfel) und 6 (Münzen kippen).
- **Mehr Unterstützung:** `ab21_a2.pbcp` als Startdatei ausgeben; Aufgabe 1 als gemeinsame Demonstration mit einem Gerät unter der Dokumentenkamera.

Alternative ohne Geräte

Ohne Geräte lässt sich das Prinzip im Emulator **nachspielen**: Die Datei `ab21_ohne_geraet.pbcp` ersetzt den Sensor durch zwei Variablen `neigung_x` und `neigung_y`. Solange eine Joystick-Richtung gedrückt ist, „kippt“ das Programm in kleinen Schritten weiter (−1 bis 1, wie g). Die Kugel rollt mit `neigung` × −5 – genau wie mit dem echten Sensor. MITTE (Leertaste) legt das Gerät wieder „flach“.



Vorschlag für die Stunde: Aufgabe 1 als Demonstration mit einem einzigen Gerät der Lehrkraft (oder als Tabelle mit den Werten oben ausgeben und deuten lassen), Aufgabe 2–4 mit dieser Datei als Vorlage: Die Schülerinnen und Schüler bauen Rand, Loch und Spielende ein. Wichtig in der

Sicherung: Mit dem Joystick bleibt die „Neigung“ stehen, wenn man loslässt – wie beim echten Kippen. Später, wenn Geräte da sind, werden die beiden Variablen einfach durch den Sensor-Block ersetzt.

Projekt: Schatzjagd

Überblick

Lernziel	Die Schülerinnen und Schüler bauen ein vollständiges Spiel schrittweise auf. Sie zerlegen es in Funktionen, speichern Positionen und Messwerte in Listen, verbinden Bedingungen mit „und“/„oder“ und erweitern das fertige Spiel selbstständig.
Dauer	3 Unterrichtsstunden à 45 Minuten, ein Arbeitsblatt (Doppelseite) pro Stunde: **AB 22a** Spielfeld, Held und Lava · **AB 22b** Geist und Schätze · **AB 22c** Zeiten und eigenes Spiel
Voraussetzungen	Sprites und Kollisionen (AB 15–17), Funktionen (AB 18), Listen (AB 19/20)
Material	PBC+ Studio (Emulator genügt, Geräte optional), Bilder `hintergrund_jagd.png`, `held.png`, `muenze.png`, `geist.png` aus dem Material-Paket; Lösungsdateien auf Anfrage: `ab22a_a1.pbcp`, `ab22a_a3.pbcp`, `ab22a_a4.pbcp`, `ab22b_a2.pbcp`, `ab22b_a4.pbcp`, `ab22b_a5.pbcp`, `ab22c_a2.pbcp` (fertiges Spiel), `ab22c_a3.pbcp`
Buch	Kapitel 20 – Projekt: Schatzjagd (Schritt 1–4 entsprechen genau den Arbeitsblättern)

Warum drei Blätter? Jede Stunde endet mit einem lauffähigen, gespeicherten Zwischenstand. Wer eine Stunde fehlt oder nicht fertig wird, startet in der nächsten Stunde mit der Lösungsdatei des vorigen Blatts (`ab22a_a3.pbcp` für 22b, `ab22b_a4.pbcp` für 22c).

Vorbereitung durch die Lehrkraft: Die vier Bilder bereitstellen. Klären, wo die Schülerinnen und Schüler ihre Datei `schatzjagd.pbcp` zwischen den Stunden speichern (Download-Ordner, Netzlaufwerk, USB-Stick). Die `.pbcp`-Datei enthält die Sprites, aber **nicht die Bilder**: Die Bilder liegen im Browser. Wer an einem anderen Rechner weiterarbeitet, muss die vier Bilder dort erneut hochladen.

Stundenverlauf

Stunde 1 – AB 22a: Spielfeld, Held und Lava

Zeit	Phase	Inhalt	Sozialform
5 min	Einstieg	Fertiges Spiel `ab22c_a2.pbcp` vorspielen. Gemeinsam die Aufgaben des Spiels sammeln → Funktionsliste an die Tafel (Buch 20.1).	Plenum
5 min	Vorbereitung	Bilder hochladen, drei Sprites anlegen	Einzelarbeit

Zeit	Phase	Inhalt	Sozialform
15 min	Aufgabe 1–2	`vorbereiten`, `held_steuern`, Aufrufe unter „Beim Start“	Einzel/Partner
15 min	Aufgabe 3–4	Leben, `gefahr_pruefen`, `ende_anzeigen`	Einzel/Partner
5 min	Sicherung	Nachgedacht: Wozu Funktionen? Speichern kontrollieren.	Plenum

Stunde 2 – AB 22b: Geist und Schätze

Zeit	Phase	Inhalt	Sozialform
5 min	Einstieg	Datei öffnen. Tafel: Tabelle der Schatzplätze (Nr. 0–4). „Wie merkt sich das Spiel fünf Plätze?“	Plenum
15 min	Aufgabe 1	Geist, „oder“ in `gefahr_pruefen`	Einzel/Partner
20 min	Aufgabe 2–3	Listen, `schatz_platzieren`, `schatz_pruefen`, Schleifenbedingung, Ende	Einzel/Partner
5 min	Sicherung	Nachgedacht (Schatz Nr. 3) und: Warum das innere „falls“?	Plenum

Stunde 3 – AB 22c: Zeiten und eigenes Spiel

Zeit	Phase	Inhalt	Sozialform
5 min	Einstieg	„Bei welchem Schatz bist du am langsamsten?“ – Zeiten sammeln und anzeigen	Plenum
15 min	Aufgabe 1–2	Liste `zeiten`, Anzeige mit „für jedes Element“	Einzel/Partner
15 min	Aufgabe 3–4	Gesamtzeit, eigene Erweiterung	Einzel/Partner
10 min	Aufgabe 5	Spieltest beim Nachbarn, zwei, drei Spiele im Plenum zeigen	Partner/Plenum

Typische Fehler

Beobachtung	Ursache	Hilfe
Ein Listen-Block arbeitet mit der falschen Liste.	Neue Listen-Blöcke kommen immer mit der ersten Liste.	Im Block auf den Listennamen (▾) klicken und die richtige Liste wählen.

Beobachtung	Ursache	Hilfe
Es passiert nichts, obwohl die Funktion fertig ist.	Die Funktion wird nirgends aufgerufen.	Namen in der Kategorie **Funktionen** anklicken, Aufruf-Block einbauen.
Der Funktionsname heißt plötzlich `schatz_pr_fen`.	Umlaute (und Leerzeichen) ersetzt die IDE beim Anlegen durch `_`.	Namen ohne Umlaute: `schatz_pruefen`.
Gleich nach dem Start: „Game Over“.	Im Block „berührt Sprite“ steht zweimal `held` – ein Sprite berührt sich immer selbst, jedes Bild kostet ein Leben.	Zweites Feld auf `geist` stellen.
Der Schatz ist unsichtbar oder liegt oben links.	`schatz_platzieren` wird in `vorbereiten` nicht aufgerufen oder „zeige Sprite schatz“ fehlt.	Beide Blöcke am Ende von `vorbereiten` ergänzen.
Nach dem letzten Schatz Fehlermeldung im Terminal.	Das innere „falls nr < Länge …“ fehlt, oder `schatz_x` und `schatz_y` sind verschieden lang.	„falls“ ergänzen; Listen vergleichen.
Meldung „Endlosschleife ohne Pause“.	„warte 20 ms“ wurde in eine Funktion verschoben – die Warnung sucht nur direkt in der Schleife.	„warte 20 ms“ in die Schleife unter „Beim Start“.
Die Lava wird nicht erkannt (eigenes Spielfeld).	Lava nicht in genau Rot (255, 0, 0) oder mit weichen Kanten gemalt.	Im Bildeditor reines Rot verwenden.
Alle Zeiten stehen übereinander in einer Zeile.	„setze zeile auf zeile + 20“ fehlt oder steht außerhalb von „für jedes Element“.	Block in die Schleife ziehen.
Die erste Zeit ist viel zu groß.	„setze start auf Zeit seit Programmstart“ fehlt in `vorbereiten`.	Am Ende von `vorbereiten` ergänzen.

Differenzierung

- **Schneller fertig:** Buch Kapitel 20, Aufgaben 4–6; Kippsteuerung aus Kapitel 18 (nur `held_steuern` austauschen, nur auf dem Gerät); eigenes Spielfeld mit **Dateien → Bild zeichnen…**
- **Mehr Unterstützung:** Zu Beginn jeder Stunde die Lösungsdatei des vorigen Blatts ausgeben. In 22b zuerst nur Aufgabe 1 und 2 (Geist, Listen) bearbeiten, Aufgabe 3 gemeinsam an der Tafel.
- **Bewertung** (falls gewünscht, siehe L 06): lauffähiges Grundspiel, Anzahl und Schwierigkeit der Erweiterungen, sinnvolle Funktionsnamen, Spieltest.

Projekt: Mein eigenes Spiel

Überblick

Lernziel	Die Schülerinnen und Schüler planen ein eigenes kleines Spiel mit einem Planungsbogen, zerlegen es in Funktionen, bauen es schrittweise, dokumentieren Tests und Fehler in einem Bau-Logbuch und stellen das Ergebnis vor.
Dauer	4–6 Unterrichtsstunden (Aufgabe 1–3 und 5–6 Pflicht, 4 und 7 Differenzierung)
Voraussetzungen	Sprites und Kollisionen (AB 15–17), Funktionen (AB 18); hilfreich: Listen (AB 19/20), das Projekt Schatzjagd (AB 22)
Material	AB 23 (4 Seiten, zwei Blätter beidseitig), PBC+ Studio (Build 109 oder neuer), Bilder aus dem Material-Paket (u. a. `raumschiff.png`, `meteor.png`, `stern.png`, `held.png`, `muenze.png`, `geist.png`). Dateien auf Anfrage: `ab23_geruest.pbcp` (Spielgerüst), `ab23_beispiel.pbcp` (Beispielspiel „Meteoritenregen“). Weitere Lösungen zum Beispielspiel: `k21_*.pbcp`. Leerer Planungsbogen auch als Kopiervorlage in L 07.
Buch	Kapitel 21, Anhang B (Fehlermeldungen), Anhang A (Blockreferenz)

Vorbereitung: Das Gerüst `ab23_geruest.pbcp` und die Bilder im Netzwerk bereitstellen. Das Beispielspiel `ab23_beispiel.pbcp` einmal im Emulator starten (Bilder `raumschiff.png` und `meteor.png` vorher hochladen). Für die Präsentation einen Beamer oder einen Rechner mit großem Bildschirm einplanen. Klären Sie vorab, ob Projekte in Einzel- oder Partnerarbeit entstehen – Partnerarbeit (zwei Personen, ein Rechner, regelmäßiger Rollenwechsel „Fahrer/Lotse“) hat sich für schwächere Lerngruppen bewährt.

> PBC+ Studio speichert nur das zuletzt bearbeitete Programm im Browser. Über mehrere Stunden hinweg müssen die Schülerinnen und Schüler am Ende **jeder** Stunde mit **Dateien → Auf den Computer…** speichern – am besten mit Stufennummer im Dateinamen. Bilder müssen nach einem Rechnerwechsel erneut hochgeladen werden.

Ablauf über mehrere Stunden

Stunde 1 – Idee und Plan

Zeit	Phase	Inhalt	Sozialform
10 min	Einstieg	Beispielspiel „Meteoritenregen“ vorführen. Gemeinsam herausarbeiten: Figuren, Steuerung, Regeln („Wenn …, dann …“), Ende.	Plenum

Zeit	Phase	Inhalt	Sozialform
10 min	Aufgabe 1	Ideenbörse lesen, Idee wählen. Kasten „Das gehört in jedes Spiel“ besprechen.	Einzel/Partner
20 min	Aufgabe 2	Planungsbogen ausfüllen, Skizze zeichnen	Einzel/Partner
5 min	Abschluss	Wer ist fertig? Reihenfolge der Planbesprechungen festlegen.	Plenum

Stunde 2 – Plan besprechen, Gerüst und Stufe 1

Zeit	Phase	Inhalt	Sozialform
5 min	Einstieg	Am Beispiel zeigen: Hauptprogramm mit leeren Funktionen (Gerüst), Stufe 1 = Steuerung. Logbuch-Eintrag vormachen.	Plenum
35 min	Bauen	Planbesprechungen (je 3–4 min, siehe Beratung), parallel: Gerüst laden, Bilder hochladen, Sprites anlegen, Steuerung bauen, erster Logbuch-Eintrag	Einzel/Partner
5 min	Sicherung	Speichern! Kurze Runde: „Was läuft schon?“	Plenum

Stunde 3 und 4 – Bauen und testen

Zeit	Phase	Inhalt	Sozialform
5 min	Impuls	Je ein kurzer Input nach Bedarf, z. B. „Warum verschwindet mein Text?“, „Mit ausgeben Werte ansehen“, „Fehlersuche in vier Schritten“ (Buch 21.4)	Plenum
35 min	Bauen	Regeln nacheinander als Stufen einbauen, testen, Logbuch führen. Lehrkraft berät (siehe unten).	Einzel/Partner
5 min	Sicherung	Speichern. Eine Gruppe zeigt einen gefundenen Fehler und die Lösung.	Plenum

Stunde 5 – Testen lassen und Feinschliff (bei 4 Stunden: in Stunde 4 integrieren)

Zeit	Phase	Inhalt	Sozialform
15 min	Aufgabe 4	Spiele tauschen, gegenseitig testen, stumm beobachten	Partner
25 min	Feinschliff	Verbesserungen aus dem Test, Kann-Liste, Präsentation vorbereiten (Aufgabe 5)	Einzel/Partner
5 min	Abschluss	Speichern, Reihenfolge der Präsentationen	Plenum

Stunde 6 – Präsentation

Zeit	Phase	Inhalt	Sozialform
35 min	Aufgabe 5/6	Präsentationen (etwa 3 min plus 1 min Rückmeldung je Projekt). Zuhörende füllen „Zwei Sterne und ein Wunsch“ aus.	Plenum
10 min	Rückblick	Rückblick-Fragen und „Das habe ich gelernt“ auf dem AB; Gespräch: Was war beim Planen hilfreich?	Einzel, Plenum

Bei großen Klassen die Präsentationen als „Spielemesse“ organisieren: Die Hälfte der Gruppen bleibt am Rechner und stellt vor, die andere Hälfte geht herum und testet; nach 15 Minuten wird gewechselt.

Beratung

Planbesprechung (Stunde 1/2, je 3–4 Minuten). Bevor gebaut wird, sieht die Lehrkraft jeden Planungsbogen an und unterschreibt. Leitfragen:

- *Ist das Spiel klein genug?* Faustregel: höchstens drei Sprites und vier bis sechs Regeln in der Muss-Zeile. Alles andere wandert in **Kann**.
- *Sind die Regeln als „Wenn …, dann …“ formuliert?* Unklare Regeln („der Gegner ist schlau“) konkretisieren lassen: „Wenn x vom Gegner kleiner als x vom Held ist, dann …“.
- *Gibt es ein Ende?* Ohne Ende fehlt die Schleifenbedingung.
- *Passen die Funktionen zu den Regeln?* Meist: `vorbereiten`, `steuern`, eine Funktion je Regelgruppe, `spiel_ende`.
- *Gibt es die nötigen Bilder?* Eigene Bilder mit **Dateien → Bild zeichnen…** kosten Zeit – das als Kann-Ziel einplanen.

Während des Bauens. Bewährt hat sich eine Hilfe-Reihenfolge: erst Logbuch und Checkliste auf Seite 3 („Wenn etwas nicht klappt“), dann Buch Anhang B, dann ein Mitschüler, erst dann die Lehrkraft. Fragen Sie bei Hilferufen zuerst: „Was soll passieren, was passiert stattdessen?“ – also die Logbuch-Spalten. Häufig finden die Schülerinnen und Schüler den Fehler schon beim Beantworten.

Wenn eine Gruppe nicht vorankommt: Plan verkleinern (eine Regel streichen), mit dem Gerüst neu anfangen oder das Beispielspiel als Basis nehmen und abwandeln (andere Bilder, andere Regeln). Auch das ist ein vollwertiges Projekt.

Bewertung

Die Kriterien und das Bewertungsraster für die Projekte AB 22 und AB 23 stehen in **L 06 (Bewertung)**. Für AB 23 lassen sich die vier Projektphasen direkt auf die Unterlagen der Schülerinnen und Schüler beziehen:

Bereich	Grundlage
Planung	Planungsbogen (Seite 2): vollständig, Regeln als „Wenn …, dann …“, sinnvolle Muss/Kann-Aufteilung
Umsetzung	Programm: läuft, erfüllt den Kasten „Das gehört in jedes Spiel“, Funktionen mit sprechenden Namen
Testen und Fehlersuche	Bau-Logbuch (Seite 3): regelmäßige Einträge, beschriebener Fehler mit Lösung
Präsentation und Rückmeldung	Vorstellung (Aufgabe 5), Qualität der eigenen Rückmeldungen (Aufgabe 6)

Bewerten Sie den **Prozess** mindestens so stark wie das fertige Spiel: Ein kleines, sauber geplantes und getestetes Spiel ist mehr wert als ein großes, unfertiges. Bei Partnerarbeit die Rollen im Logbuch festhalten lassen.

Typische Fehler

Beobachtung	Ursache	Hilfe
Text (Punkte, „Start“) verschwindet sofort wieder.	Jede Sprite-Bewegung malt den ganzen Bildschirm neu (Hintergrund + Sprites).	Text auf den Canvas schreiben („Text auf Canvas“) oder erst anzeigen, wenn die Sprites versteckt sind; Leben z. B. mit LEDs anzeigen (wie im Beispielspiel).
Bei einer Berührung sind sofort alle Leben weg.	Die Sprites berühren sich mehrere Schleifendurchläufe lang.	Nach dem Treffer ein Sprite versetzen (neuer Gegner, Held zurück zum Start). Buch 21.4.
Meldung „Endlosschleife ohne Pause“, obwohl ein „warte“ da ist	Das „warte“ steht in einer Funktion, die in der Schleife aufgerufen wird; die Prüfung schaut nicht in Funktionen hinein.	„warte 20 ms“ direkt in die Schleife legen – oder „OK“ wählen.
Meldung „Funktionsaufruf ohne Funktion“, im Aufruf steht „(?)“	Funktion umbenannt oder gelöscht	Im Aufruf die richtige Funktion wählen.
„Sprite … berührt Sprite …“ ist immer wahr.	Frisch geholt steht in beiden Feldern dasselbe Sprite.	Zweites Feld umstellen.
Auf dem Bildschirm steht „?“ statt „ä“.	Die Bildschirmschrift kennt keine Umlaute.	ae, oe, ue, ss schreiben.

Beobachtung	Ursache	Hilfe
Programm vom Vortag ist weg.	Nicht gespeichert, anderer Rechner	Am Stundenende speichern; Dateien im Klassenordner sammeln.

Differenzierung

- **Schneller fertig:** Aufgabe 7 (Version 2); Kann-Liste umsetzen; eigene Bilder mit **Dateien → Bild zeichnen…**; ein zweites Level (z. B. nach 10 Punkten anderer Hintergrund und höheres Tempo); Buch Kapitel 21, Aufgaben 3–5.
- **Mehr Unterstützung:** Mit dem Gerüst starten; eine Idee aus der Ideenbörse wählen, zu der es ein Buchkapitel gibt (Fang den Apfel ≈ Kapitel 13/21, Lava-Labyrinth ≈ Kapitel 13, Reaktionstest ≈ Kapitel 19); das Beispielspiel abwandeln statt neu bauen; Partnerarbeit mit klarer Rollenverteilung.
- **Ohne Geräte:** Das ganze Projekt läuft im Emulator. Der Beschleunigungssensor liefert dort feste Werte (Gerät liegt flach) – die Idee „Kipp-Kugel“ ist daher nur mit echten PicoBoys sinnvoll.

Bewertung

Kriterien, Punkteraster und Bögen für die beiden Projekte der Reihe: AB 22 „Schatzjagd“ (Nachbau mit Erweiterung, 3 Stunden) und AB 23 „Mein eigenes Spiel“ (freies Projekt, 4–6 Stunden).

Grundsätze

- **Prozess und Produkt.** Bewertet wird nicht nur das fertige Spiel, sondern auch Planung, Testen und Fehlersuche. Ein Spiel, das am Ende noch einen Fehler hat, kann sehr gut sein, wenn die Gruppe den Fehler gefunden, beschrieben und eingegrenzt hat.
- **Transparenz.** Teilen Sie das Raster zu Beginn aus und besprechen Sie es: Struktur und Testen zählen, nicht nur die Spielidee.
- **Gruppenarbeit.** Anteile sichtbar machen: Der Planungsbogen (L 07) enthält eine Aufgabenverteilung; bei der Präsentation erklärt jedes Gruppenmitglied einen Teil des Programms.
- **Belege.** `.pbcp`-Datei (Zwischenstände je Stunde speichern lassen), Planungsbogen mit Testprotokoll, Ihre Beobachtungen, Präsentation.
- **Anpassen.** Gewichtung und Notenschlüssel sind Vorschläge. Richten Sie sich nach den Vorgaben Ihrer Schule und Ihrer Fachkonferenz. Ohne Noten taugen die Stufen als Rückmeldung in Worten: 3 = „sicher“, 2 = „überwiegend“, 1 = „in Ansätzen“, 0 = „noch nicht“.

Die Kriterien

Kriterium	Woran Sie es erkennen	AB 22	AB 23
Planung	Planungsbogen ausgefüllt: Idee, Regeln, Steuerung, Figuren, Variablen/Listen, Funktionen; Arbeitsschritte geplant	–	✓
Funktionsfähigkeit	Programm startet ohne Fehlermeldung, Steuerung funktioniert, Spielende (gewonnen/verloren) wird erreicht	✓	✓
Struktur	Funktionen mit sinnvollen Namen, keine unnötig doppelten Blöcke, Listen statt vieler Einzelvariablen, verständliche Variablennamen, ggf. Kommentare	✓	✓
Kreativität / Erweiterung	eigene Ideen: AB 22 – Erweiterungen des Grundspiels; AB 23 – eigene Spielidee, Gestaltung, Spielbarkeit	✓	✓
Testen und Fehlersuche	Testprotokoll geführt, Fehler beschrieben und behoben, schrittweise gebaut und getestet	✓	✓
Präsentation	Spiel vorgeführt, Aufbau des Programms erklärt, Fragen beantwortet	✓ (kurz)	✓

Raster AB 22: Projekt Schatzjagd

Jedes Kriterium wird mit 0–3 Punkten bewertet und mit dem angegebenen Faktor multipliziert. **Höchstpunktzahl: 30.**

Kriterium	3 Punkte	2 Punkte	1 Punkt	0 Punkte
Funktionsfähigkeit ×3	Das Grundspiel läuft vollständig: Steuerung, Schätze einsammeln, Lava und Geist, Spielende mit Zeit. Keine Fehlermeldung.	Das Spiel läuft, ein Teil fehlt oder funktioniert nicht zuverlässig (z. B. Spielende).	Nur einzelne Teile laufen (z. B. Steuerung und ein Schatz).	Das Programm startet nicht oder bricht sofort mit Fehler ab.

Kriterium	3 Punkte	2 Punkte	1 Punkt	0 Punkte
Struktur ×3	Teile des Spiels stehen in benannten Funktionen; Schatz-Positionen und Zeiten stehen in Listen und werden mit Schleifen verarbeitet.	Funktionen und Listen werden verwendet, aber teilweise umgangen (doppelte Blöcke, Einzelvariablen).	Programm steht überwiegend in einem langen Block unter „Beim Start“.	Nicht nachvollziehbar, viele lose Blöcke.
Erweiterung ×2	Mindestens zwei eigene, funktionierende Erweiterungen (z. B. mehr Schätze, Level, Leben, Ton, Bestenliste).	Eine eigene Erweiterung funktioniert.	Erweiterung begonnen, funktioniert noch nicht.	Keine Erweiterung.
Testen und Fehlersuche ×1	Mindestens drei Fehler im Testprotokoll beschrieben, Ursache und Lösung notiert.	Fehler notiert, Lösung teilweise.	Testprotokoll lückenhaft.	Kein Testprotokoll.
Präsentation ×1	Spiel vorgeführt und eine Funktion verständlich erklärt.	Spiel vorgeführt, Erklärung lückenhaft.	Nur vorgeführt.	Keine Präsentation.

Rechnung: (Funktionsfähigkeit × 3) + (Struktur × 3) + (Erweiterung × 2) + Testen + Präsentation = höchstens 9 + 9 + 6 + 3 + 3 = **30 Punkte**.

Vorschlag für einen Notenschlüssel

Note (Anteil der Punkte)	1 (ab 87 %)	2 (ab 73 %)	3 (ab 59 %)	4 (ab 45 %)	5 (ab 20 %)	6
AB 22 (30 P.)	26	22	18	14	6	0–5
AB 23 (45 P.)	39	33	27	21	9	0–8

Raster AB 23: Mein eigenes Spiel

Höchstpunktzahl: 45. Jedes Kriterium 0–3 Punkte, mal Faktor.

Kriterium	3 Punkte	2 Punkte	1 Punkt	0 Punkte
Planung ×2	Planungsbogen vollständig: Idee, Regeln, Steuerung, Skizze, Figuren, Variablen/Listen, Funktionen, Aufgabenverteilung. Plan wurde genutzt und bei Bedarf angepasst.	Planungsbogen weitgehend ausgefüllt, ein Bereich fehlt.	Nur Idee und Skizze.	Keine Planung.
Funktionsfähigkeit ×3	Spiel ist von Anfang bis Ende spielbar, hat ein klares Ziel und Ende, keine Fehlermeldung.	Spielbar, aber ein Teil fehlt oder hakt (z. B. kein Spielende, Kollision ungenau).	Nur Teile laufen; kein durchgehendes Spiel.	Startet nicht.
Struktur ×3	Mindestens zwei sinnvoll benannte Funktionen; Listen oder Variablen passend eingesetzt; keine lose herumliegenden Blöcke; Namen verständlich.	Eine Funktion oder Listen/Variablen teilweise passend; Struktur erkennbar.	Alles in einem langen Block, aber lesbar.	Unübersichtlich, nicht nachvollziehbar.
Kreativität ×2	Eigene Spielidee, liebevoll gestaltet (Bilder, Farben, Töne), macht Spaß und ist nicht zu leicht oder zu schwer.	Eigene Idee mit Gestaltung, Spielbarkeit mit Schwächen.	Weitgehend Nachbau eines Beispiels mit kleinen Änderungen.	Unveränderter Nachbau.
Testen und Fehlersuche ×2	Testprotokoll mit mindestens drei Fehlern (Beobachtung – Ursache – Lösung); schrittweise gebaut; andere haben das Spiel getestet.	Fehler notiert, Lösungen teilweise.	Testprotokoll lückenhaft.	Kein Testprotokoll.
Präsentation ×3	Spiel vorgeführt, Aufbau (Funktionen, Variablen/Listen) verständlich erklärt, eine Schwierigkeit und ihre Lösung geschildert, Fragen beantwortet.	Vorgeführt und erklärt, kleine Lücken.	Nur vorgeführt, kaum Erklärung.	Keine Präsentation.

Rechnung: Planung × 2 + Funktionsfähigkeit × 3 + Struktur × 3 + Kreativität × 2 + Testen × 2 + Präsentation × 3 = höchstens 6 + 9 + 9 + 6 + 6 + 9 = **45 Punkte**.

Bewertungsbogen Projekt

Name(n): _____ Klasse: _____ Projekt: ☐ AB 22 Schatzjagd ☐ AB 23 eigenes Spiel

Kriterium	0	1	2	3	Faktor	Punkte
Planung *(nur AB 23)*	☐	☐	☐	☐	× 2	
Funktionsfähigkeit	☐	☐	☐	☐	× 3	
Struktur (Funktionen, Listen, Namen)	☐	☐	☐	☐	× 3	
Erweiterung (AB 22) / Kreativität (AB 23)	☐	☐	☐	☐	× 2	
Testen und Fehlersuche	☐	☐	☐	☐	AB 22: × 1 · AB 23: × 2	
Präsentation	☐	☐	☐	☐	AB 22: × 1 · AB 23: × 3	
Summe					von 30 / 45	

Das ist besonders gelungen:

Daran kannst du noch arbeiten:

Beobachtungen während der Arbeit (Datum, Notiz):

Mein Projekt – so schätze ich mich ein

Name: _____ Klasse: _____ Datum: _____

Kreuze an, was für dich und dein Projekt stimmt.

	stimmt genau	stimmt eher	stimmt eher nicht	stimmt nicht
Ich habe mein Spiel geplant, bevor ich mit dem Bauen angefangen habe.	☐	☐	☐	☐
Mein Spiel startet ohne Fehlermeldung.	☐	☐	☐	☐
Man kann mein Spiel gewinnen oder verlieren.	☐	☐	☐	☐
Ich habe Teile meines Programms in Funktionen aufgeteilt und ihnen gute Namen gegeben.	☐	☐	☐	☐
Ich habe Listen oder Variablen sinnvoll benutzt.	☐	☐	☐	☐
Ich habe nach jedem Schritt getestet.	☐	☐	☐	☐
Wenn etwas nicht ging, habe ich den Fehler selbst gesucht, bevor ich gefragt habe.	☐	☐	☐	☐
Im Spiel stecken eigene Ideen von mir.	☐	☐	☐	☐
Ich kann erklären, wie mein Programm funktioniert.	☐	☐	☐	☐
Ich habe gut mit meiner Partnerin / meinem Partner zusammengearbeitet.	☐	☐	☐	☐

Darauf bin ich stolz:

Dieser Fehler hat mich am meisten beschäftigt – und so habe ich ihn gelöst:

Das würde ich beim nächsten Mal anders machen:

Kopiervorlagen

Druckfertige Vorlagen für den Unterricht. Jede Vorlage steht auf einer eigenen Seite und kann einzeln kopiert werden.

Vorlage	Inhalt	Einsatz
A	Bildschirm-Koordinaten 240 × 280, (0, 0) oben links	AB 02–03, 09, 14–17: Positionen planen, Zeichnungen entwerfen. Auch als Folie.
B	Turtle-Winkelscheibe: Richtung 0 = oben, 90 = rechts	AB 11–13: Drehungen und Richtungen; ausschneiden und mit einer Musterbeutelklammer einen Zeiger befestigen
C	Turtle-Koordinaten, (0, 0) in der Mitte, y nach oben	AB 11–13: Position setzen, Figuren planen
D	Farbtabelle der benannten Farben mit Rot/Grün/Blau-Werten	AB 03, 14, 17: Farben mischen, Kollisionsfarben in Bildern
E	Projekt-Planungsbogen (2 Seiten)	AB 23, auch AB 22 und Mini-Projekte
F	Tasten-Übersicht Gerät und Emulator	ab AB 07; als Tischkarte laminieren

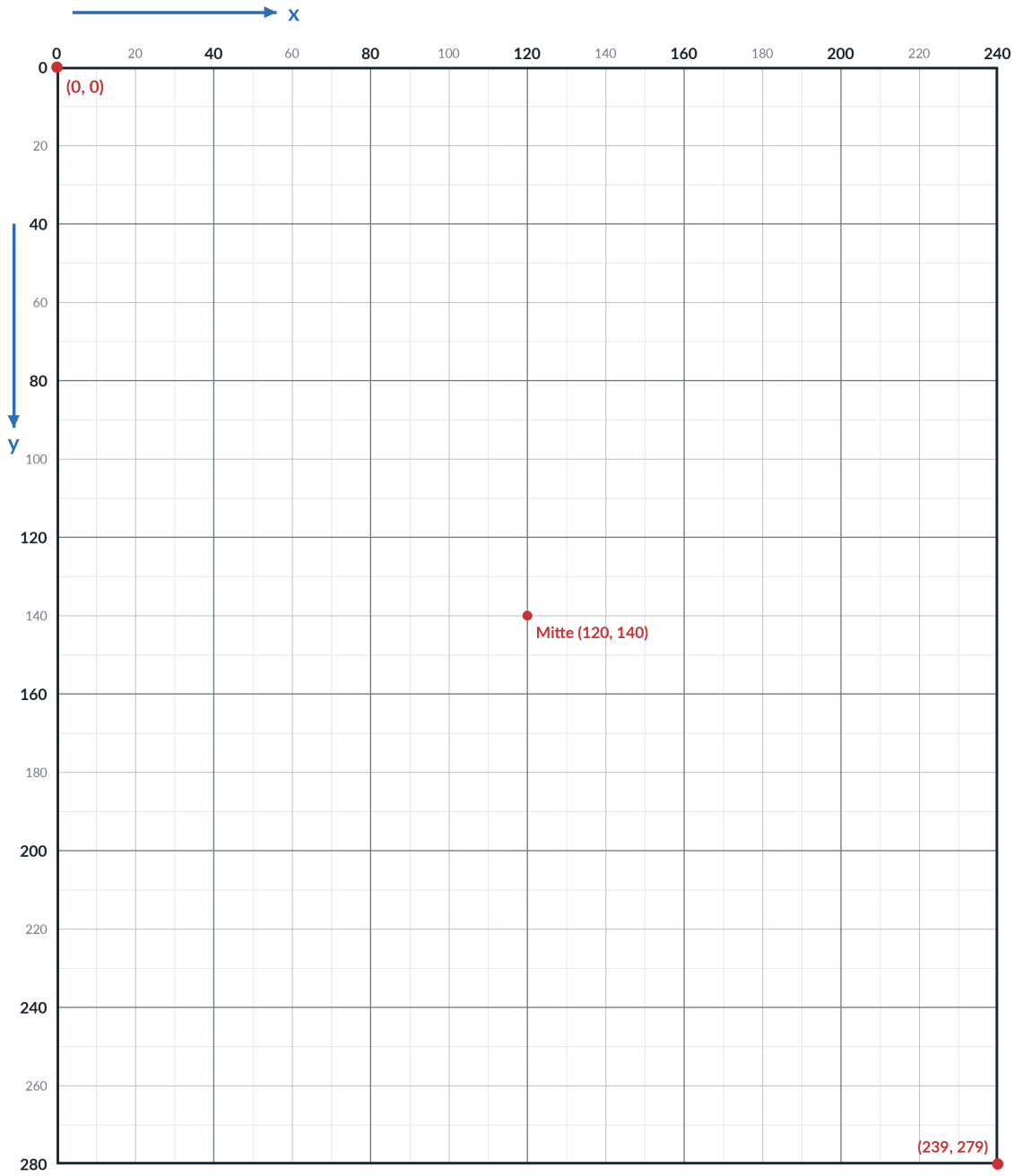
Hinweise zu den Vorlagen

- **A und C** sind maßstabsgetreu gezeichnet: ein Kästchen = 10 Pixel. Schülerinnen und Schüler können darauf Formen einzeichnen und die Koordinaten direkt ablesen. Achten Sie beim Wechsel von A zu C auf die unterschiedliche Richtung von y.
- **B** zeigt die Richtungen so, wie PBC+ Studio sie verwendet (wie ein Kompass, im Uhrzeigersinn): „Richtung 0“ = oben, 90 = rechts, 180 = unten, 270 = links. „rechts drehen“ dreht im Uhrzeigersinn, „links drehen“ gegen den Uhrzeigersinn. Die Turtle startet nach oben blickend in der Mitte.
- **D:** Die Werte gelten für „Farbe nach Namen“. Der Bildschirm zeigt 65 536 Farben; selbst gemischte Farben werden leicht gerundet.
- Die Abbildungen A–C erzeugt das Skript `werkzeuge/abbildungen_lehrer.py`.

KOPIERVORLAGE A

Der Bildschirm: 240 × 280 Pixel

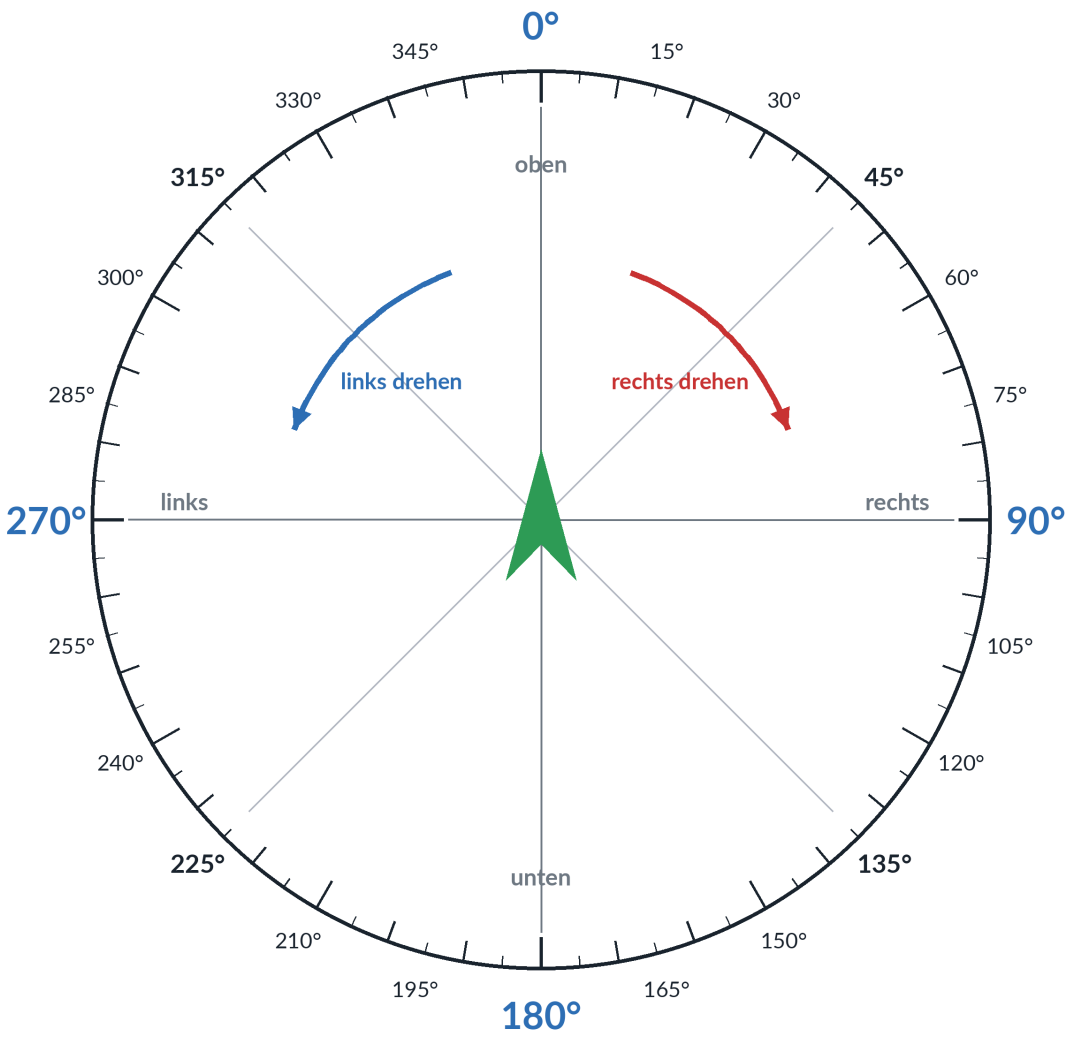
(0, 0) ist oben links. x wächst nach rechts, y wächst nach **unten**. Ein Kästchen = 10 Pixel.



KOPIERVORLAGE B

Turtle: Richtungen und Drehungen

„Richtung 0“ zeigt nach oben, 90 nach rechts. „rechts drehen“ dreht im Uhrzeigersinn, „links drehen“ dagegen.



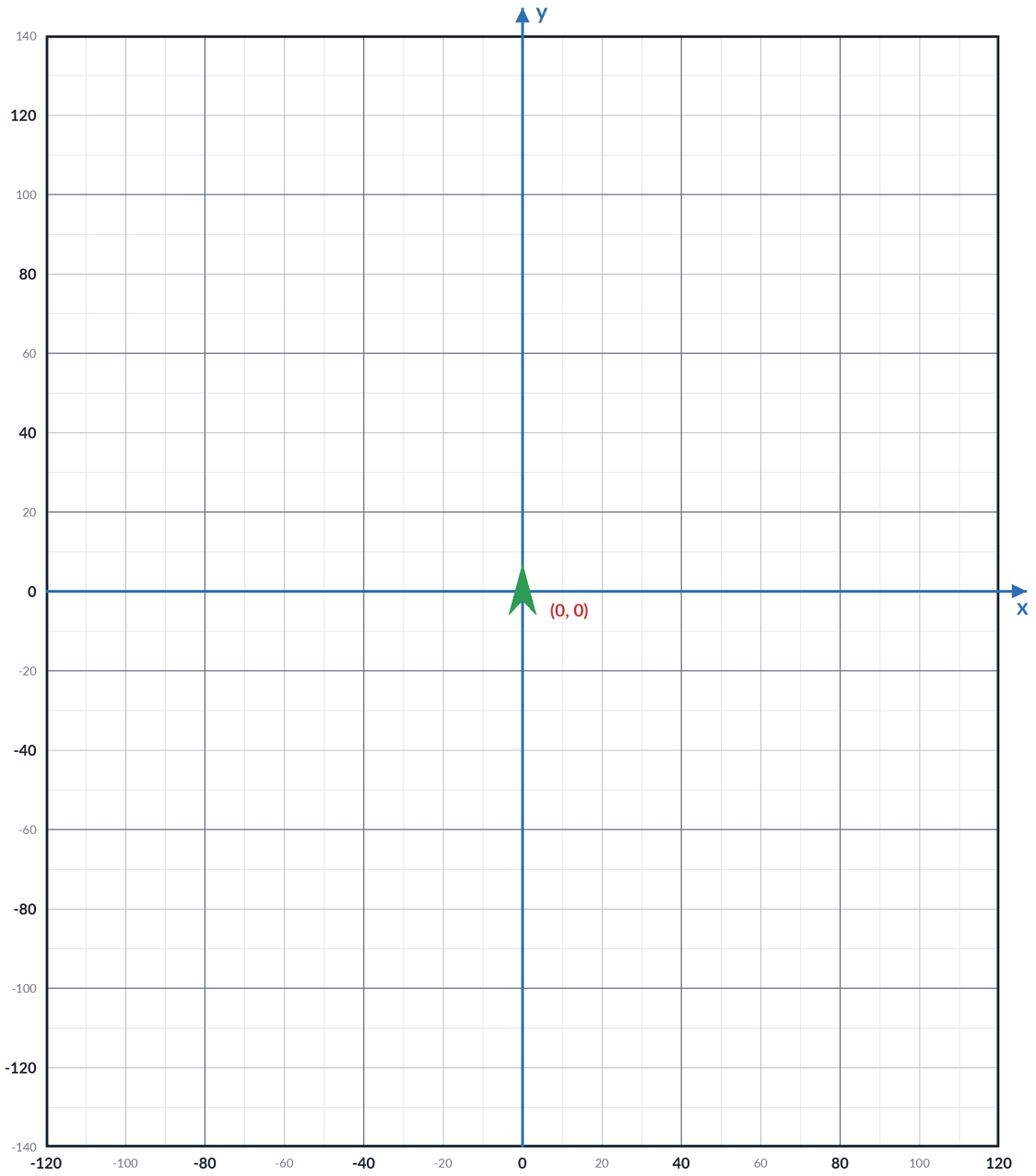
Drehwinkel für regelmäßige Vielecke: Die Turtle dreht an jeder Ecke um 360° geteilt durch die Zahl der Ecken.

Figur	Dreieck	Quadrat	Fünfeck	Sechseck	Achteck	Stern (5 Zacken)
Ecken × Drehung	3 × 120°	4 × 90°	5 × 72°	6 × 60°	8 × 45°	5 × 144°

KOPIERVORLAGE C

Turtle-Koordinaten

(0, 0) ist in der Mitte. x wächst nach rechts, y wächst nach **oben**. Die Turtle startet hier und blickt nach oben. Ein Kästchen = 10 Pixel.



KOPIERVORLAGE D

Farben

Der Block „Farbe nach Namen“ kennt diese elf Farben. Mit dem Block „Farbe rot … grün … blau …“ mischst du jede Farbe selbst: Jeder Wert geht von **0** (gar nichts) bis **255** (ganz viel).

Farbe	Name im Block	Rot	Grün	Blau	Hinweis
	weiß	255	255	255	alles voll
	schwarz	0	0	0	alles aus
	rot	255	0	0	
	grün	0	255	0	
	blau	0	0	255	
	gelb	255	255	0	rot + grün
	cyan	0	255	255	grün + blau
	magenta	255	0	255	rot + blau
	grau	128	128	128	alles halb
	orange	255	165	0	ungefähr
	lila	120	0	120	genau

Zum Ausprobieren – gemischte Farben:

Farbe	Idee	Rot	Grün	Blau	Farbe	Idee	Rot	Grün	Blau
	braun	139	69	19		rosa	255	192	203
	dunkelgrün	0	100	0		himmelblau	135	206	235
	dunkelblau	0	0	128		gold	255	215	0

> **Für Lehrkräfte – Kollisionsfarben in Bildern:** „berührt Farbe“ vergleicht exakt. Wer Wände oder Lava in einem Bildprogramm malt, verwendet genau die Werte aus der oberen Tabelle und zeichnet ohne Kantenglättung. Am sichersten sind rot, grün, blau, gelb, cyan, magenta, weiß und schwarz.

Projekt-Planungsbogen: Mein Spiel

Namen: _____ Klasse: _____ Datum: _____

1. Name und Idee

Mein Spiel heißt: _____

Darum geht es (2–3 Sätze):

2. So sieht es aus

Bildschirm 240 × 280 – zeichne deine Figuren ein.

3. Figuren und Bilder

Name (ohne Umlaute)	Bilddatei / Aussehen	Was macht sie?

4. Steuerung

Taste	Was passiert?
HOCH / RUNTER	
LINKS / RECHTS	
MITTE	
A	
B	

5. Regeln

So bekommt man Punkte:	
So gewinnt man:	
So verliert man:	
Töne / LEDs:	

6. Variablen und Listen

Name (ohne Umlaute)	Zahl, Text oder Liste?	Was wird gespeichert?	Startwert

7. Funktionen

Name der Funktion	Was macht sie?	Wer baut sie?	fertig
			☐
			☐
			☐
			☐

8. Arbeitsplan

Stunde	Das wollen wir schaffen	geschafft
1		☐
2		☐
3		☐
4		☐
5		☐

9. Testprotokoll

Das habe ich beobachtet	Das war die Ursache	So habe ich es gelöst

Tipp: Speichere am Ende jeder Stunde mit **Dateien → Auf den Computer…** und gib der Datei einen Namen mit der Stunde, z. B. `spiel_std2.pbcp`.

KOPIERVORLAGE F

Tasten am PicoBoy und im Emulator



Am Gerät	Im Block	Im Emulator (Tastatur)
Joystick nach oben	HOCH	Pfeil nach oben ↑
Joystick nach unten	RUNTER	Pfeil nach unten ↓
Joystick nach links	LINKS	Pfeil nach links ←
Joystick nach rechts	RECHTS	Pfeil nach rechts →
Joystick drücken	MITTE	Leertaste
Taste A	A	Taste **A**
Taste B	B	Taste **B**

Im Emulator kannst du die Knöpfe auch mit der Maus anklicken.

Welcher Block? „wird … gedrückt?“ ist wahr, **solange** die Taste unten ist – gut zum Bewegen. „wurde … seit der letzten Abfrage gedrückt?“ meldet jeden Druck **genau einmal** – gut zum Zählen oder Umschalten.

Startmenü auf dem Gerät	
Programm auswählen	Joystick hoch / runter
Programm starten	A, B oder MITTE
Zurück ins Menü	Programm endet – oder **RES** auf der Rückseite

In PBC+ Studio	
Starten / anhalten	▶ **Testen** / ■ **Stop**
Einen Block löschen	anklicken, **Entf**
Rückgängig	**Strg + Z**
Sichern	**Dateien → Auf den Computer…**