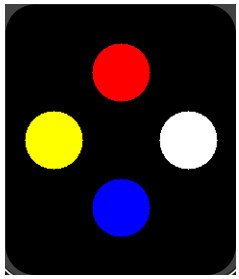
Name: _____ Klasse: _____ Datum: _____

Listen 2: Merk dir die Folge

Ziel: Du programmierst das Merkspiel „Simon sagt“. Der PicoBoy lässt farbige Felder in einer Folge aufleuchten, du spielst sie mit dem Joystick nach. In jeder Runde wird die Folge um eine Farbe länger. Die Folge steckt in einer **Liste**.

So geht das Spiel: Vier Felder: oben rot, rechts grün, unten blau, links gelb. Jedes Feld hat eine **Nummer** von 0 bis 3 und einen eigenen Ton. Drückst du den Joystick in eine Richtung, leuchtet das Feld dort weiß auf.

Vorbereitung: Wähle **Emulator** und beginne ein neues Programm (**Dateien → Neu**). Speichere nach jeder Aufgabe (**Dateien → Auf den Computer…**).

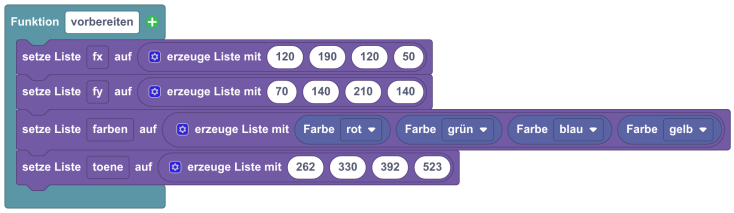


Rechts (grün) leuchtet gerade auf.

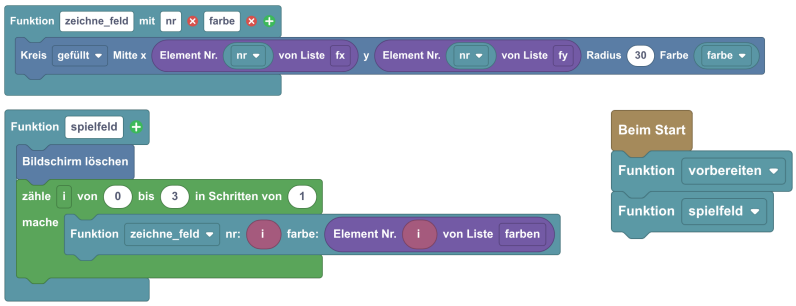
Stunde 1: Spielfeld und Folge

Aufgabe 1 ⭐ Das Spielfeld

Lege die Listen `fx`, `fy`, `farben` und `toene` an (Kategorie **Listen**, „+ neue Liste…“). Baue die Funktion `vorbereiten`. Die Lücken im Block „erzeuge Liste mit“ stellst du mit dem Zahnrad ein.



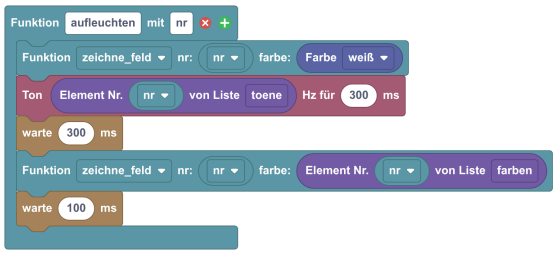
Baue dann die Funktion `zeichne_feld` mit zwei Parametern (grünes **+**, umbenennen in `nr` und `farbe`) und die Funktion `spielfeld`:



Rufe in „Beim Start“ `vorbereiten` und `spielfeld` auf (rechts) und teste.

Welche Werte holt `zeichne_feld` für das Feld Nr. 3 aus den Listen? x = _____ y = _____ Farbe = _____

Aufgabe 2 ⭐ Aufleuchten



Die Funktion `aufleuchten` malt das Feld weiß, spielt seinen Ton und malt es wieder in seiner Farbe.

Baue die Funktion. Teste sie, indem du unter „Beim Start“ zwei Aufrufe anhängst: „aufleuchten nr: 0“ und „aufleuchten nr: 2“.



Warum steht nach dem Ton-Block noch „warte 300 ms“?

Aufgabe 3 ⭐ Die Folge

Die Folge, die du dir merken musst, ist eine Liste von Feld-Nummern. Diese Folge bedeutet: unten, oben, links, links.

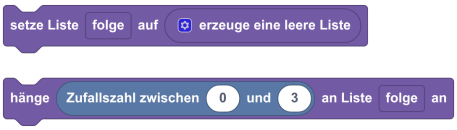
Welche Farben leuchten bei der Folge **[1, 1, 3, 0]**?

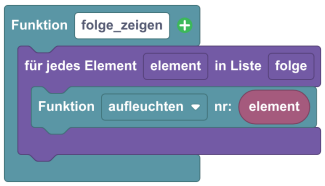
Liste folge

2	0	3	3
Nr. 0	Nr. 1	Nr. 2	Nr. 3

0 = oben (rot) 1 = rechts (grün)
2 = unten (blau) 3 = links (gelb)

Lege die Liste `folge` an. Baue die Funktion `folge_zeigen`. Ersetze die Test-Aufrufe aus Aufgabe 2 durch diese Blöcke und hänge darunter „warte 1000 ms“ und den Aufruf `folge_zeigen`:

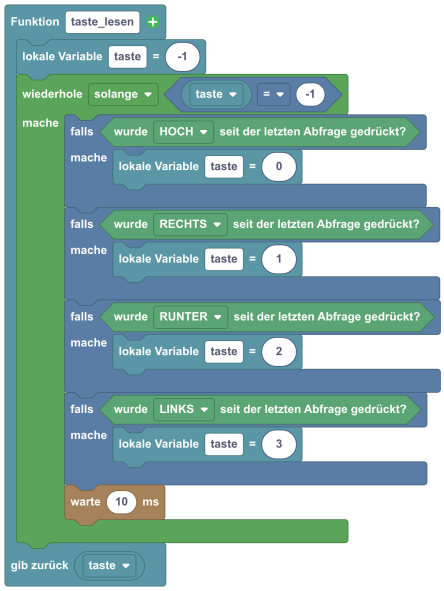




Hänge die Zufallszahl **dreimal** an. Teste mehrmals: Ist die Folge jedes Mal anders?

Stunde 2: Nachspielen

Aufgabe 4 ⭐ Den Joystick lesen

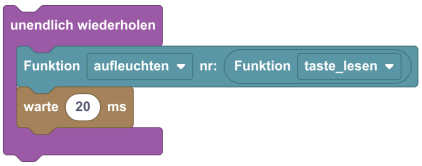


`taste_lesen` ist eine **Funktion mit Rückgabe**: „+ neue Funktion mit Rückgabe…“. Sie wartet, bis du den Joystick in eine Richtung drückst, und gibt die Nummer dieser Richtung zurück.

Die **lokale Variable** `taste` legst du unter `taste_lesen()` mit „+ neue lokale Variable…“ an. Den zweiten bis fünften Block „lokale Variable taste =“ bekommst du mit Rechtsklick → **Kopieren**.

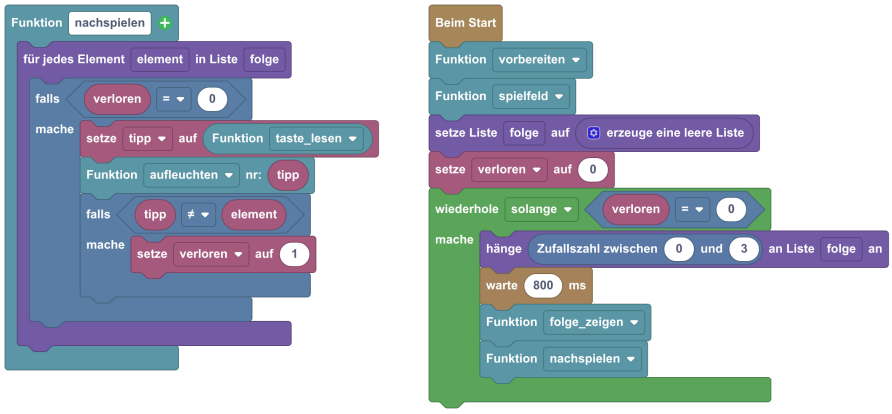
`taste` startet mit −1. Warum −1 und nicht 0?

Test: Ersetze in „Beim Start“ alles unter `spielfeld` durch diese Schleife. Jetzt leuchtet das Feld auf, in dessen Richtung du drückst.



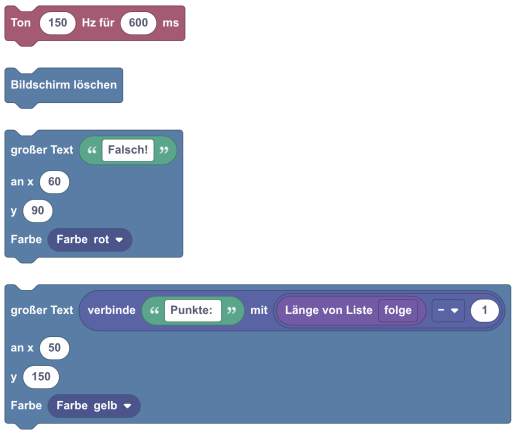
Aufgabe 5 ⭐⭐ Das Spiel

Lege die Variablen `verloren` und `tipp` an. Baue die Funktion `nachspielen` und ändere „Beim Start“ wie im Bild unten:



Erkläre: Warum wird in `nachspielen` geprüft, ob `verloren` noch 0 ist?

Aufgabe 6 ⭐⭐ Game over



Hänge diese Blöcke **unter** die „solange“-Schleife. Warum steht dort „Länge von Liste folge − 1“ und nicht einfach die Länge?

Mein Rekord: _____ Punkte

Aufgabe 7 ⭐⭐⭐ Immer schneller

Mit jeder Runde soll das Aufleuchten kürzer werden: 400 ms minus 30 ms pro Element der Folge, aber nie kürzer als 100 ms. Lege eine Variable `tempo` an und setze sie am Anfang jeder Runde auf

„Maximum von 100 und (400 − Länge von Liste folge × 30)“

(Block „Minimum / Maximum“ aus **Mathe**). Benutze `tempo` in `aufleuchten` statt der beiden 300. Weitere Ideen: ein Fehler-Ton, eine Bestenliste wie in AB 19, ein fünftes Feld für die Taste A.

Das habe ich gelernt:

- ☐ Ich kann Werte, die zusammengehören, in mehreren Listen mit gleicher Nummer speichern.
- ☐ Ich kann mit „Element Nr. …“ einen bestimmten Wert aus einer Liste holen.
- ☐ Ich kann eine Liste mit „hänge … an“ wachsen lassen und ihre Länge benutzen.
- ☐ Ich kann eine Funktion mit Rückgabe und einer lokalen Variable bauen.

Mehr dazu im Buch: Kapitel 15 – Listen (und Kapitel 14 – Funktionen)