Name: _____ Klasse: _____ Datum: _____

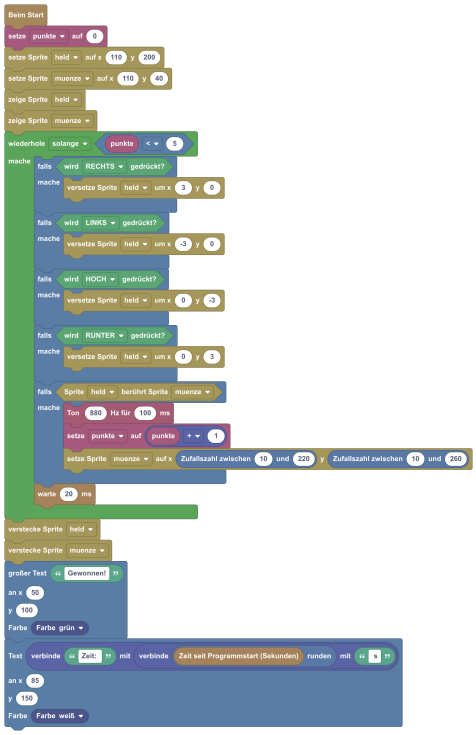
Ordnung mit Funktionen

Ziel: Du räumst ein langes Programm auf. Du teilst es in **Funktionen** mit guten Namen auf. Danach liest sich „Beim Start“ wie ein Inhaltsverzeichnis.

Vorbereitung

1. Wähle oben rechts **Emulator**. Lade die Bilder `held.png` und `muenze.png` hoch: **Dateien → Bilder verwalten…**
2. Öffne die Datei `ab18_vorlage.pbcp`: **Dateien → Vom Computer…**

Aufgabe 1 ⭐ Was macht welcher Teil?



Teste das Spiel mit ▶ **Testen**. Es ist das Münzspiel aus AB 16.

Das Programm hat vier Teile. Schreibe zu jedem Teil, welche Blöcke dazu gehören (z. B. „die ersten 5 Blöcke“).

Vorbereiten (Startwerte, Sprites):

Steuern (Held bewegen):

Münze prüfen (Treffer, Punkte):

Ende zeigen (Gewonnen, Zeit):

Aufgabe 2 ⭐ Die erste Funktion: `steuern`



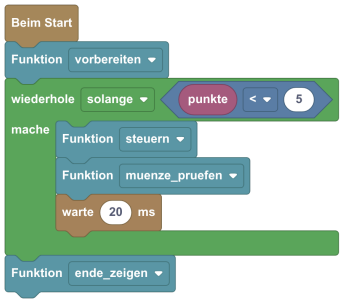
1. Kategorie **Funktionen** → **„+ neue Funktion…“**, Name `steuern`.
2. Ziehe den ersten „falls wird RECHTS gedrückt?“-Block in die neue Funktion. **Achtung:** Alle Blöcke darunter kommen mit!
3. Ziehe den Block „falls Sprite held berührt Sprite muenze“ (mit „warte“ darunter) zurück in die Schleife.
4. Klicke in der Kategorie **Funktionen** auf `steuern()` und setze den Aufruf oben in die Schleife:



5. Teste: Läuft der Held noch?

Aufgabe 3 ⭐ Alles aufräumen

Baue genauso die Funktionen `vorbereiten`, `muenze_pruefen` und `ende_zeigen`. Am Ende soll „Beim Start“ so aussehen:



Teste nach **jeder** neuen Funktion. Speichere dein Programm: **Dateien → Auf den Computer…**

Ziehe den Aufruf `steuern` kurz aus der Schleife heraus und teste. Was passiert? Warum?

Aufgabe 4 ⭐⭐ Eine Funktion mit Parameter: `piep`



Lege eine Funktion `piep` an. Klicke auf das grüne **+**: Ein Parameter `x` erscheint. Klicke hinein und nenne ihn `hoehe`. Den Block `hoehe` findest du in der Kategorie **Funktionen** unter `piep(hoehe)`. Ersetze in `muenze_pruefen` den Ton-Block durch den Aufruf (unten links) und spiele in `ende_zeigen` einen tiefen Piep mit 440.

Aufgabe 5 ⭐⭐⭐ Ein Geist kommt dazu

Lade `geist.png` hoch und lege ein Sprite `geist` an. Schreibe eine Funktion `geist_bewegen`: Der Geist schwebt hin und her (wie in Kapitel 13). Berührt er den Helden, muss der Held zurück an den Start. Welche Funktionen musst du noch ändern?

Das habe ich gelernt:

- ☐ Ich kann eine Funktion anlegen und Blöcke hineinziehen.
- ☐ Ich kann eine Funktion aufrufen.
- ☐ Ich kann einer Funktion einen Parameter geben.
- ☐ Ich finde gute Namen, die sagen, was eine Funktion tut.

Mehr dazu im Buch: Kapitel 14 – Funktionen