Name: _____ Klasse: _____ Datum: _____

Listen 3: Sternenregen

Ziel: Du lässt viele Sterne gleichzeitig fallen. Ihre Positionen stehen in Listen: erst in **parallelen Listen**, dann als **Listen in Listen**. Du erzeugst Listen mit einer **List Comprehension** und entfernst Sterne sicher aus einer Liste.

Aufgabe 1 ⭐ Drei Sterne, zwei Listen

ab17_start.py

```
 1 import pbc
 2 import time
 3
 4 xs = [40, 120, 200]
 5 ys = [0, 60, 120]
 6
 7 pbc.clear()
 8 while True:
 9     for i in range(len(xs)):
10         pbc.fill_rect(xs[i], ys[i], 4, 4, pbc.BLACK)
11         ys[i] += 3
12         if ys[i] > 275:
13             ys[i] = 0
14         pbc.fill_rect(xs[i], ys[i], 4, 4, pbc.WHITE)
15     time.sleep_ms(30)
```

Tippe ab (oder öffne `ab17_start.py`) und teste.

Stern Nummer `i` steht bei `xs[i]`, `ys[i]`. Listen, deren Elemente mit gleichem Index zusammengehören, heißen **parallele Listen**.

Was tun Zeile 10 und 14?

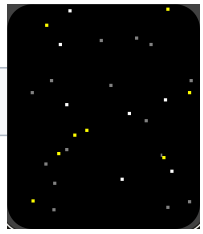
Was tun Zeile 12 und 13?

Füge einen vierten Stern hinzu. Was musst du ändern?

Aufgabe 2 ⭐ 30 Sterne in einer Zeile

Eine **List Comprehension** erzeugt eine ganze Liste nach einer Regel:

```
xs = [random.randint(0, 236) for i in range(30)]
```



„Die Liste aus 30 Zufallszahlen von 0 bis 236." Erzeuge so `xs` und `ys` (0 bis 275) für 30 Sterne. Denk an `import random`.

Gib jedem Stern außerdem ein eigenes Tempo von 1 bis 4 in einer dritten Liste `tempo` und benutze `ys[i] += tempo[i]`. Schnelle Sterne wirken näher: Färbe sie mit `FARBEN = [pbc.BLACK, pbc.GRAY, pbc.GRAY, pbc.WHITE, pbc.YELLOW]` und `FARBEN[tempo[i]]`.

Aufgabe 3 ⭐⭐ Ein Stern – eine Liste

Statt drei Listen bekommt jeder Stern eine kleine Liste `[x, y, tempo]`:

```
sterne = [[random.randint(0, 236), random.randint(0, 275), random.randint(1, 4)]
          for i in range(30)]
```

Schreibe die Spielschleife um: `for stern in sterne:` und `stern[1] += stern[2]`. Was ist `sterne[0][2]`? _____

Aufgabe 4 ⭐⭐ Gelandet ist gelandet

Ein Stern, der unten ankommt, soll jetzt aus der Liste **verschwinden**. Erst ein Test in der Konsole. Was gibt das Programm aus, und was sollte es ausgeben?

_____ statt _____

```
1 sterne_y = [280, 290, 100, 300]
2 for y in sterne_y:
3     if y > 275:
4         sterne_y.remove(y)
5 print(sterne_y)
```

Wird während der Schleife entfernt, rücken die Elemente nach vorn, und eines wird übersprungen. Abhilfe: über eine **Kopie** laufen, `for stern in sterne[:]:`, und mit `sterne.remove(stern)` im Original entfernen. Baue das ein, zähle die gelandeten Sterne und zeige `len(sterne)` an. Mit Taste A kommt oben ein neuer Stern dazu (`append`). Die Schleife läuft `while len(sterne) > 0:`.

Warum wird beim Durchlaufen der Kopie kein Stern übersprungen?

Aufgabe 5 ⭐⭐⭐ Ausweichen!

Unten fliegt ein Raumschiff (ein Rechteck), das du mit dem Joystick nach links und rechts steuerst. Sterne, die unten ankommen, starten wieder oben. Berührt ein Stern das Schiff, ist das Spiel vorbei. Zeige, wie viele Sekunden du überlebt hast (`time.ticks_ms`). Plane zuerst: Wann berührt ein 4 × 4 Pixel großer Stern bei (x, y) ein Schiff bei (`schiff_x`, `SCHIFF_Y`) mit 20 × 12 Pixeln? Schreibe die Bedingung auf:

Das habe ich gelernt:

- ☐ Ich kann viele Objekte mit parallelen Listen oder Listen in Listen bewegen.
- ☐ Ich kann eine Liste mit einer List Comprehension erzeugen.
- ☐ Ich weiß, warum ich beim Entfernen über eine Kopie laufe.

Mehr dazu im Buch: Kapitel 14 – Listen im Spiel