Name: _____ Klasse: _____ Datum: _____

Projekt Weltraumspiel 3: Punkte und Highscore

Ziel: Dein Weltraumspiel bekommt Punkte, einen Endbildschirm und einen Highscore, der in einer Datei gespeichert wird. Danach baust du eine eigene Erweiterung ein.

Vorbereitung: Öffne `ab23c_start.py` (das Spiel aus AB 23b mit Leben und „GAME OVER“). Speichere jede funktionierende Fassung unter neuem Namen, z. B. `weltraum_1.py`, `weltraum_2.py` …

Aufgabe 1 ⭐ Punkte für jedes Ausweichen

Für jeden Meteor, der unten am Schiff vorbeifliegt, gibt es einen Punkt.



1. Ersetze `zeige_leben` durch diese Funktion und passe alle Aufrufe an:

```
def zeige_stand(punkte, leben):
    himmel.fill_rect(0, 0, 240, 20, pbc.BLACK)
    himmel.text12(f"Punkte: {punkte}", 10, 4, pbc.WHITE)
    himmel.text12(f"Leben: {leben}", 170, 4, pbc.WHITE)
    pbc.show(himmel)
```

2. Lege `punkte = 0` an. Wo zählst du den Punkt? Im Zweig `elif meteor.y > pbc.HEIGHT:` – ergänze dort `punkte += 1` und `zeige_stand(punkte, leben)`.
3. Schreibe die Punkte auch in den GAME-OVER-Kasten (`text18`).

Aufgabe 2 ⭐ Den Rekord speichern

Ergänze die Lücken und schreibe beide Funktionen über `# Hintergrund`. Oben brauchst du `import os` und die Konstante `DATEI = "weltraum_rekord.txt"`.

```
def lade_rekord():
    if DATEI in os.listdir():
        with open(DATEI, "____") as f:
            return ______(f.read())
    return 0


def speichere_rekord(wert):
    with open(DATEI, "____") as f:
        f.write(______(wert))
```

Am Spielende, nach dem Verstecken der Sprites, lädst du den alten Rekord mit `rekord = lade_rekord()`. Ist `punkte` größer, wird `rekord = punkte` und mit `speichere_rekord(rekord)` gespeichert. Zeige `rekord` im Endbildschirm an (`text12`).

Aufgabe 3 ⭐ Testen

Spiele zweimal hintereinander. Welcher Rekord steht beim zweiten Spiel da?

_____

Lade jetzt die Seite PBC+ Studio im Browser neu und spiele noch einmal. Was passiert mit dem Rekord, und warum?

_____

_____

Aufgabe 4 ⭐⭐ Deine Erweiterung

Wähle eine Idee (oder eine eigene), plane sie in zwei, drei Schritten und baue sie ein:

- **Bonusstern** (`stern.png`): fällt langsam, bringt 5 Punkte, wenn du ihn fängst.
- **Immer schneller:** Alle 10 Punkte fallen die Meteore schneller (Tipp: `tempo = TEMPO_METEOR + punkte // 10`).
- **Neustart:** Nach „GAME OVER“ startet Taste A ein neues Spiel.
- **Ufo** (`ufo.png`): fliegt waagrecht über den Bildschirm; eine Berührung kostet ein Leben.
- **Licht:** Die RGB-LED leuchtet bei einem Treffer kurz rot.

Meine Idee: _____

Schritt 1: _____

Schritt 2: _____

Schritt 3: _____

Aufgabe 5 ⭐⭐⭐ Zwei Erweiterungen

Baue eine zweite Idee aus Aufgabe 4 ein. Teste nach jedem Schritt. Was hat nicht sofort geklappt, und wie hast du den Fehler gefunden?

_____

_____

_____

Das habe ich gelernt:

☐ Ich kann Punkte und Leben gemeinsam anzeigen.

☐ Ich kann einen Highscore in einer Datei speichern und wieder laden.

☐ Ich weiß, dass Dateien im Emulator nur bis zum Neuladen der Seite bleiben.

☐ Ich kann ein Spiel planvoll in kleinen Schritten erweitern.

Mehr dazu im Buch: Kapitel 15 – Dateien, Kapitel 21 – Projekt: Ein Spiel Schritt für Schritt