

## Ziel der Stunde - Code aufräumen

Die Schüler lernen,

- **doppelten Code zu erkennen,**
- gemeinsame Logik **an einer zentralen Stelle zu bündeln,**
- bestehende Programme **übersichtlicher und wartbarer** zu machen,
- mehrere parallele Endlosschleifen sinnvoll **zusammenzufassen.**

👉 Schwerpunkt ist **nicht** neues Gameplay, sondern **sauberer Code.**

## Lehrplanbezug

- Algorithmisches Denken
- Strukturierung von Programmen
- Wiederverwendung von Code  
Problemlösen & Optimierung
- Erste Prinzipien von Software-Qualität

*(Informatikdidaktisch: Einführung in Refactoring ohne Fachbegriff-Zwang)*

## Material

- Scratch-Projekt vom vorherigen Teil
- Schüler-Arbeitsblatt „Teil 5“

## Grundidee dieser Stunde

Bis hierhin haben die Schüler funktionierenden Code gebaut.  
Jetzt geht es um die Frage:

**„Was passiert eigentlich mehrmals – und warum?“**

Die Schüler sollen erkennen:

- gleicher Code an mehreren Stellen = **unnötig kompliziert**
- Änderungen werden fehleranfällig
- zentrale Logik macht Programme robuster

## Didaktischer Hinweis

Diese Stunde ist **bewusst anspruchsvoll**, aber:

- sie trennt „funktioniert irgendwie“ von „**sauber programmiert**“
- sie zeigt, dass Informatik **nicht nur Neues bauen**, sondern auch **Verbessern & Vereinfachen** ist