

# Pinball-Projekt für Programmieren und Algorithmen

Institut für Experimentelle Teilchenphysik

Prof. Dr. T. Ferber, Dr. B. Maier, Dr. A. Poenicke  
→ [Veranstaltung in Ilias](#)

WS2023/24 – Pinball-Projekt

---

Entwickeln Sie in einem Team aus 2-3 Mitgliedern einen Flipper (“Pinball”), den Sie am Ende des Semesters im Team in einer 10-15 minütigen mündlichen Präsentation vorstellen werden.

Für die Prüfung gilt:

- Fragen werden gestellt zu den von Ihnen gewählten Code-Elementen, Funktionen, ...
- Es wird nur auf Bestehen/Nichtbestehen geprüft
- Bestehen auf individueller Basis (d.h. es könnten auch nur 2 von 3 Gruppenmitgliedern bestehen)
- Eine Chance zur Wiederholung ~1-2 Wochen später
- Die drei besten Flipper gewinnen Preise
- Bei Krankheit: ärztliches Attest & Terminverschiebung
- **Wichtig: Bitte teilen Sie uns sofort mit, wenn sich Ihre Gruppe im Laufe des Semesters auflösen sollte.**

Starten Sie von dieser Vorlage [https://gitlab.kit.edu/benedikt.maier/pinball\\_start](https://gitlab.kit.edu/benedikt.maier/pinball_start) und modifizieren bzw. erweitern Sie sie entsprechend.

Es ist empfohlen, den Code modular zu gestalten, also auch neue Dateien zu kreieren, in denen Funktionen, Klassen, etc. definiert sind, die dann eingebunden werden.

- **Der Code muss folgende Eigenschaften besitzen**

- Gravitation ✓
- Kollisionen mit Wand ✓
- Abbruchbedingung (z.B. “Loch im Boden”) ✓
- Kollisionen mit verschiedenen Elementen: Kreis, Rechteck + ein sich bewegendes Element (kein rotierender Kreis) nach Wahl
- Energieverlust oder Beschleunigung bei Kollisionen (kein Luftwiderstand)
- Zwei Flipperhebel & eine Abschussvorrichtung
- Muss mit mehreren Bällen zurechtkommen und einen neuen Ball bei bestimmten Ereignissen spawnen ✓
- Punkte-Tracking innerhalb eines Spiels und Highscore-Tracking über Spiele hinweg ✓
- Bonus: Irgendeinen Penalty oder Tilt, Sound, ...

- **Außerdem gilt zu berücksichtigen:**

- Basieren Sie Ihren Code auf `pygame` und die in Ihrer `pua.yml` Umgebung installierten Bibliotheken **ausser `pymunk` und `collide`-Funktionen von `pygame`.**
  - Implementieren Sie Kollisionen zwischen Objekten selbst (d.h. ohne Verwendung von oben genannten `pymunk` oder `collide`-Funktionen aus `pygame`)
  - Programmieren Sie Objekt-orientiert!
  - Dies ist ein Semester-langes Projekt. Fangen Sie damit nicht erst drei Wochen vor der Prüfung an.
  - Nutzen Sie das Projekt-tutorium, um von uns Tips und Hilfestellung zu erhalten.
-