

# Game Programming

Game Programming ist die faszinierende Disziplin, in der du lernst, wie du die Logik und Technik hinter Spielen entwickelst. Von der Programmierung interaktiver Spielmechaniken bis zur Optimierung von Codes - die Quests hier führen dich durch die Schritte, um die technische Essenz der Spiele zu verstehen und zu beherrschen.

- [Lerne die Coding Grundlagen!](#)
- [Code eine Spielmechanik!](#)
- [Code ein Spielsystem!](#)

# Lerne die Coding Grundlagen!

Lerne die wichtigsten Grundlagen des Programmierens mit Javascript Hero:

<https://www.jshero.net>

Nachdem du die Grundkenntnisse erlangt hast ,erhältst du eine auf dich abgestimmte Coding Herausforderung von der Laborleitung. Wenn du diese erfolgreich einem Senior oder der Laborleitung präsentiert hast ist die Quest erfolgreich abgeschlossen.

**\*\*Diese Quest muss absolviert werden bevor mit anderen Game Engines als Construct 3 gearbeitet werden kann!\*\***

# Code eine Spielmechanik!

**Dauer:** ~~

**Das lernst du:**

- erstellen und umsetzen Spielmechaniken

## Was sind Spielmechaniken?

Am besten lässt sich das mit einem Beispiel erklären:

In Super Mario kann Mario die Gegner besiegen in dem er ihnen auf den Kopf springt, dies wäre eine Spielmechanik. Mario kann aber auch verschiedene Fähigkeiten erlangen um z.B. Gegner mit einem Feuerball abzuschießen. Dies wäre eine weitere Mechanik. Mechaniken lassen sich oft gut mit einem Verb beschreiben z.B. springen, sammeln, schießen, zielen, kombinieren usw. Im Spiellabor gibt es Karten mit Spielmechaniken, diese können dir dabei helfen herauszufinden welche Spielmechanik du deinem Spiel hinzufügen willst.

- 1. Erstelle ein neues Construct Projekt mit dem Titel "Spielmechaniken" und denke dir entweder eine eigene Mechanik aus oder nimm eine Karte aus dem Mechaniken Fach des Spiellabor Vorrats.**
- 2. Überlege dir wie du die gewählte Mechanik in Construct umsetzen könntest.**
- 3. Wenn deine Mechanik funktioniert präsentiere sie einem Senior oder der Laborleitung.**

# Code ein Spielsystem!

**Dauer:** ~~

**Das lernst du:**

- was sind Spielsysteme
- wie setzt man Spielsysteme in Construct um

## Was sind Spielsysteme?

In vielen Spielen findet man immer wieder ähnliche Abläufe oder Verhaltensweisen, zum Beispiel gibt es bei Rollenspielen meistens ein Inventar mit dem der Spieler seine gefundenen Gegenstände sortieren und einsetzen kann. Genauso hat nahezu jedes Spiel eine bestimmte Art und Weise wie es mit Auseinandersetzungen/Kampf zwischen Spieler und Spielwelt oder auch Spielern untereinander umgeht. Bei diesen beiden Beispielen handelt es sich um Systeme, ein Inventar bzw. Kampfsystem. Spielsysteme stellen unsere Spielobjekte und Mechaniken in einen sinnvollen Zusammenhang schaffen Klarheit, was, wann und wie mit unseren Spielobjekten passiert bzw. passieren kann.

Beispiel:

- Ich möchte das die Spielfigur bestimmte Gegenstände in der Welt aufsammeln kann und sie in einem Beutel verstaut. Der Spieler soll Gegenstände aus dem Beutel dann auf Knopfdruck ansehen und verwenden können. Für den beschriebenen Fall wäre ein Inventarsystem zuständig. Hier würden wir definieren welche Gegenstände aufgesammelt werden können, wie und wann der Spieler Gegenstände aufsammeln, ansehen und verwenden kann und z.B. wie viele Gegenstände wir überhaupt in unserem Beutel haben können. Auch würden wir uns überlegen wie unser Inventar dem Spieler präsentiert wird: Wird, wie bei Minecraft, ein Teil des Inventars immer angezeigt oder gibt es dafür ein eigenes Menü ?

**Wähle eines der genannten Spielsysteme und erstelle ein Construct Projekt in dem du das gewählte System umsetzt:**

- Inventarsystem
- Rundenbasiertes Kampfsystem
- Action Kampfsystem
- Dialogsystem