



BESCHREIBUNG

Für die Interaktion mit Objekten im Spiel, wie beispielsweise das Einsammeln von Münzen, ist es notwendig, Kollisionen zu erkennen. Wir bauen die notwendigen Funktionen und ein System ein, um Kollisionen mit mehreren unterschiedlichen Objekten zu erkennen, zu unterscheiden und darauf zu reagieren.

Wir fügen hierfür unserem Spiel Münzen zum Einsammeln sowie einen Gegner hinzu. Beide sollen am Ende zufällig auf unseren Plattformen gespawnt werden.



ZIELE

Löse die folgenden Aufgaben:

- Lade folgende Assets aus dem Asset Store in dein Projekt (Der Kurs verwendet die „**GDevelop Platformer**“ Assets):
 - Münze
 - Monster
- Erstelle dynamisch je eine Münze pro Plattform. Erweitere hierfür die Funktion **createPlatform**.
- Implementiere eine Funktion **inCollisionWidth** zur Kollisionsabfrage zwischen dem einem Objekt und einer Liste von Objekten.
 - Das Objekt (hier der PC) und die Liste sollen als Eingangsparameter übergeben werden.
 - Die Funktion soll die Instanz aus der Objektliste zurückgeben, mit welcher die Kollision stattgefunden hat.
- Hole Dir alle Instanzen der Münzen (**Coin**) und speichere sie in einer Variable **Coins** ab.
- Nutze die Funktion **inCollisionWith** um zu prüfen, ob eine Kollision zwischen PC und einer Münze stattgefunden hat.
 - Die Probe soll nur ausgeführt werden, wenn **Coins** länger als 0 ist.
 - Wenn eine Kollision mit einem Objekt mit dem Typ/Name **Coin** stattgefunden hat, soll diese gelöscht werden.



Monster Erweiterung:

- Erweitere die Funktion **createPlattform** nochmal, so dass dynamisch entweder eine Münze oder ein Monster je Plattform spawned wird. **Tipp:** Du kannst den **scaleFactor** als Entscheidungskriterium verwenden.
- Hole Dir alle Instanzen der Münzen (**Monster**) und speichere sie in einer Variablen **Monsters** ab.
- Erweitere die Kollisionsabfrage um die Monster, so dass zwischen Monster und Münzen unterschieden werden kann. Tipp: Füge die beiden Listen **Coins** und **Monsters** zusammen.
 - Bei Kollision mit einem Monster soll das Level neu geladen werden.