

Grundlagen der GameMaker-Engine

Gameloop

Basis einer jeden Gameengine ist diese sogenannte Gameloop. Eine Schleife die während des Spielens endlos läuft und grob gesagt in jedem Durchlauf einen neuen Spielframe berechnet und diesen auf den Bildschirm zeichnet. In jedem Durchlauf werden u.a. alle Spielobjekte aktualisiert (z.B. die Position anhand der letzten Position und der momentanen Geschwindigkeit berechnet), Kollisionen erkannt und verarbeitet und zum Schluss das neue Bild gerendert.

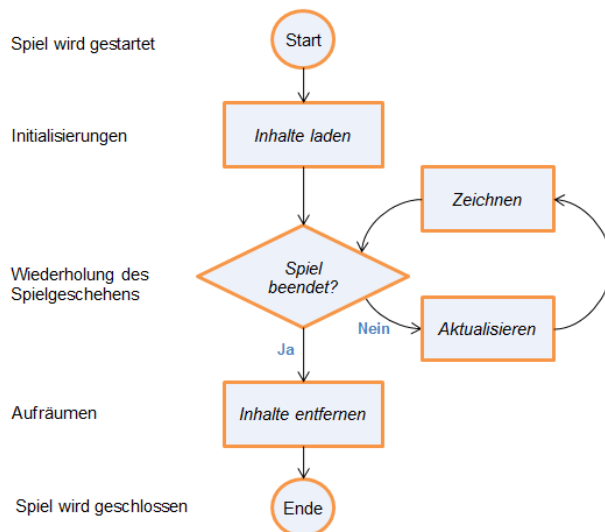


Abbildung 1 Gameloop

Sprites

Bewegliche Spielgrafiken in GMS werden Sprites genannt. Ein Sprite in GMS kann dabei aus mehreren einzelnen Bildern (Frames) bestehen. Wenn ein Sprite aus mehreren Grafiken besteht, wird es automatisch animiert, indem einfach zwischen den Einzelbildern gewechselt wird.

Kollisionsmaske

Eine wichtige Einstellung eines Sprites ist, die Kollisionsmaske. Wenn sich die Kollisionsmasken von zwei Objekten überschneiden, wird eine Kollision zwischen beiden erkannt. Es gibt hierbei verschiedene Arten:

| Art | Geschwindigkeit |
|-----------------------|-----------------|
| Rechteck ohne Drehung | Sehr schnell |
| Rechteck | Schnell |
| Ellipse | Mittel |
| Diamant | Mittel |
| Präzise | Langsam |
| Präzise pro Frame | Sehr langsam |

Man sollte sich immer gut überlegen, welche Kollisionsmaske man einem Sprite gibt. In den meisten Fällen ist eine Kollisionsmaske, die das Sprite nur ungefähr abdeckt völlig ausreichend. Nimmt man hingegen für alle Sprites *Präzise*, kann sich das sehr stark auf die Performance des Spiels auswirken und zu Ruckeln führen.

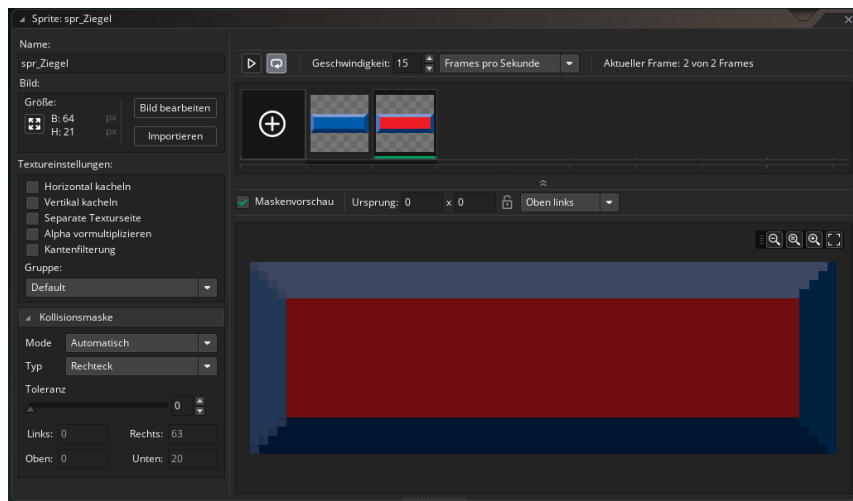


Abbildung 2 Spritfenster

Objekte

Um ein Spiel zu erstellen müssen wir Objekte definieren. In unserem Beispiel, ein Objekt für den Balken, die Ziegel und den Ball. Von diesen Objekten kann man dann Instanzen in das Spiel direkt einfügen, indem man sie einfach in einen *Raum* einfügt. Jedem Spielobjekt kann man ein Sprite zuweisen.

Jede Instanz eines Objekts verfügt auch über eigene Variablen, welche man unter *Variablendefinitionen* deklarieren kann. Viele wichtige Variablen sind schon in allen Objekten vordefiniert.

Ein paar wichtige vordefinierten Variablen:

| Name | Beschreibung |
|--------------------------------------|---|
| <i>mouse_x, mouse_y</i> ¹ | x-, bzw. y- Koordinate des Mauszeigers |
| <i>x, y</i> | Die Positionskoordinaten des Objekts |
| <i>speed</i> | Die Geschwindigkeit des Objekts |
| <i>vspeed, hspeed</i> | Die vertikale bzw. horizontale Geschwindigkeit des Objekts. Der Wert dieser Variable gibt an, um wie viele Pixel sich das Objekt im nächsten Frame vertikal bzw. horizontal verschiebt. |

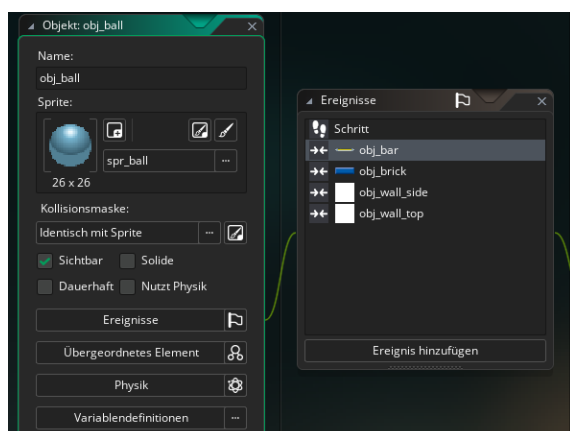


Abbildung 3 Objektfenster

¹ Ist genaugenommen keine Objektvariable sondern eine globale Variable

Ereignisse

Über Ereignisse wird das Verhalten von Objekten geregelt. Wenn man z.B. festlegen will, was passiert wenn der Ball mit der Wand kollidiert, kann man das Ereignis *Kollision mit Wand* verwenden.

Ein sehr wichtiges Ereignis ist Schritt. Dieses Ereignis wird einfach in jedem Durchlauf der Gameloop für alle Objekte aufgerufen.

Weiters gibt noch das Zeichnen-Ereignis. Dieses wird am Ende eines Durchlaufs der Gameloop aufgerufen. WICHTIG: Nur in diesem Ereignis kann man Dinge (z.B. einen Text, oder eine Anzeige) direkt auf den Bildschirm zeichnen!

Variablen

Mithilfe von Variablen kann man Daten speichern und verändern. In Gamemaker wird z.B. in der Variable *hspeed* die horizontale Geschwindigkeit eines Objekts gespeichert. Wenn man diese Variable umkehrt (d.h. das Vorzeichen ändert) dann wechselt das Objekt die horizontale Bewegungsrichtung. Die Bewegung nach Oben/ Unten bleibt gleich.

Im Objektfenster kannst du unter Variablendefinitionen auch deine eigenen Variablen definieren.

Den Wert einer Variable kann man mit dem Block *Variable Zuweisen* ändern

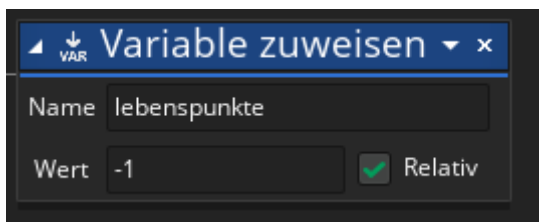


Abbildung 4 Die Lebenspunkte werden um 1 verringert

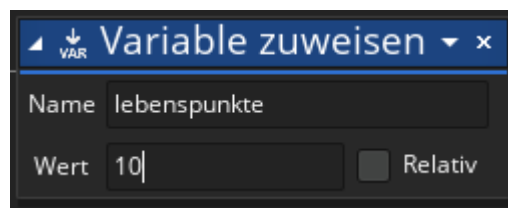


Abbildung 5 Die Lebenspunkte werden auf 10 gesetzt

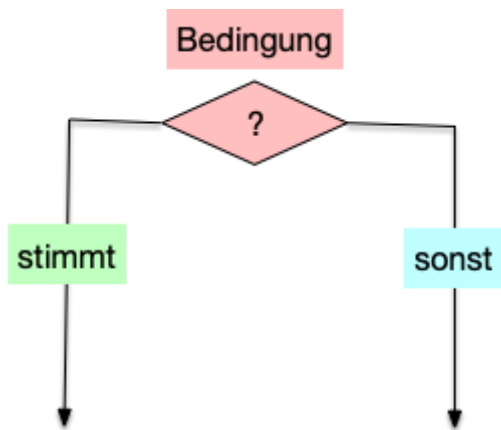
Wenn der Haken bei *Relativ* gesetzt ist, wird der Wert dazu addiert.

Kollisionen

Wenn sich die Kollisionsmasken zweier Objekte überschneiden, dann wird ein Kollisionsereignis ausgelöst.

Will man zum Beispiel, dass ein Ball, wenn er mit einer Wand kollidiert abprallt muss man in das Kollisionsereignis des Balls mit der Wand Code einfügen, dass den Ball abprallen lässt. (Dafür reicht es evtl. einfach die Geschwindigkeit umzukehren) Lässt man das Kollisionsereignis leer, passiert einfach nichts und der Ball fliegt durch die Wand hindurch.

Verzweigung



Um Dinge zu programmieren, die nur dann passieren, wenn eine bestimmte Bedingung erfüllt ist.
(z.B. nur wenn die Lebenspunkte auf 0 sind)

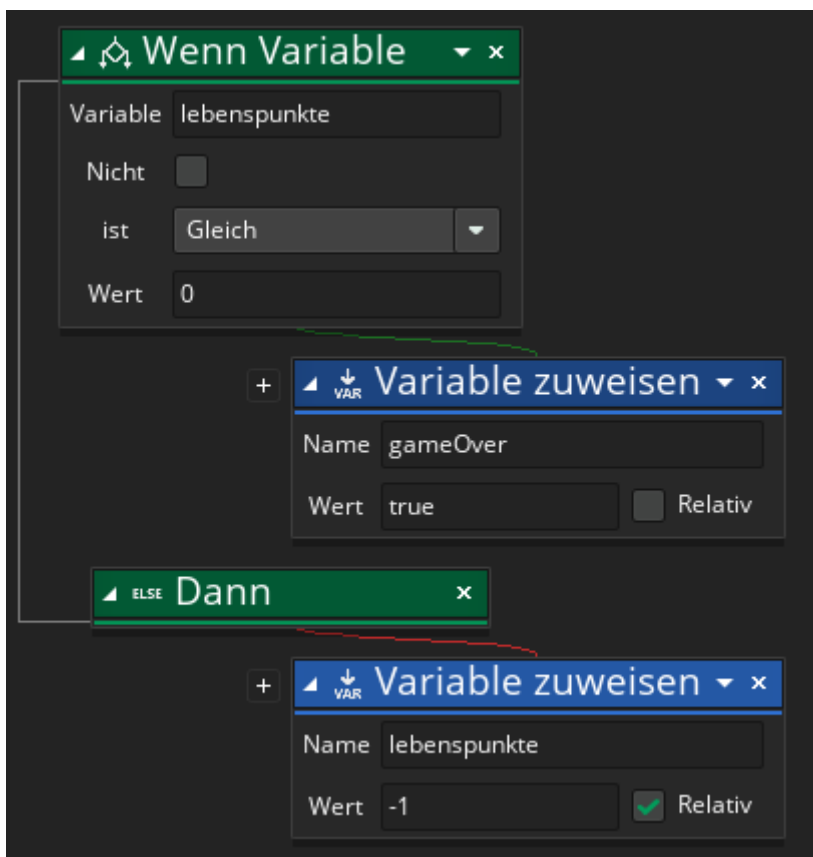


Abbildung 6 Wenn die Lebenspunkte auf 0 sind, dann wird die Variable GameOver auf true gesetzt, wenn dies nicht der Fall ist, werden die Lebenspunkte um eins reduziert

Mit dem Else-Block legt man fest, was passiert, wenn die Bedingung im Wenn nicht erfüllt ist.
(ACHTUNG: Der Else-Block wird in der aktuellen Version von Gamemaker fälschlicherweise mit *Dann* übersetzt. Das sollte eigentlich *Sonst* heißen!)