



Beschreibung

Jetzt wird es Zeit ein richtiges Level zu bauen!

Als Erstes fügen wir ein paar Objekte hinzu, um unser Level weiter auszuschnücken.

Dafür geben wir unserer Plattform ein paar zugehörige Endstücke und weisen diesen Behavior „**Plattform**“ hinzu.

Damit die Objekte alle schon sauber angelegt sind, passen wir das Grid an, an dem sich unsere Objekte ausrichten können und orientieren die Größe und die Ausrichtung unserer Objekte an dem Grid, damit diese auch einheitlich sind.

Stellt sicher, dass die Kacheln in eurem Grid in den Einstellungen dieselbe Höhe und Breite wie eure Assets haben, damit diese sauber aneinander ausgerichtet werden.

Ihr könnt das Grid zwischendrin auch wieder deaktivieren, um eure Elemente freier zu platzieren.

Damit unser Frosch noch höher springen kann, um die Plattformen zu erreichen, gehen wir in die Einstellungen von unseren Behaviours und passen dort die Werte des Jump Behaviours so an, dass der Frosch auch die Plattformen erreichen kann. In unserem Beispiel ändern wir dabei den Wert für den **Jumpspeed und Fallingspeed auf 800**, ihr könnt auch mit den Werten herumexperimentieren, um so das Level einfacher oder schwerer zu gestalten.

Spawnpoints

Damit unser Level noch weiter an Schwierigkeit gewinnt und für die Spielenden herausfordernder wird, fügen wir in diesem Abschnitt **Spawnpoints** hinzu.

Als Erstes setzen wir Wasser in unsere Spielszene hinzu, damit unsere Spielfigur etwas zum Hineinfallen hat. Das Wasser Asset legen wir als „**Tiled Sprite**“ Objekt an, damit dieses beim in die Länge ziehen weiter fortgesetzt wird.

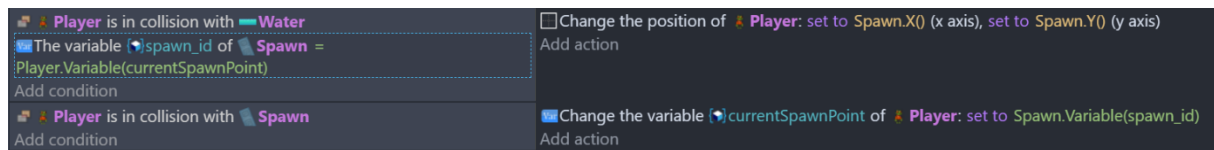
Dann fügen wir eine Collision hinzu: Wenn die Spielfigur das Wasser berührt, soll sich die Position des Charakters wieder auf den Start bewegen.

Dafür geben wir die Koordinaten von der Anfangsposition unseres Frosches ein, diese können wir aus dem Properties Panel ablesen. In diesem Fall sind die Startkoordinaten bei *X auf 90 und bei Y auf 630*.

Im nächsten Schritt legen wir noch weitere Punkte als Spawnpoints fest. Dafür legt ihr ein neues Objekt an, mit einem Asset eurer Wahl und nennt dieses „**SpawnPoint**“ und platziert dieses in eurem Level.

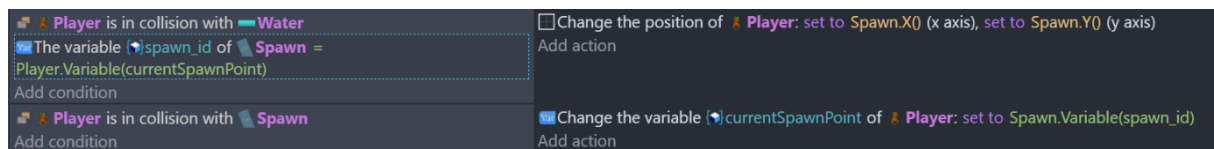


Danach wechselt ihr wieder auf das Event Sheet und fügt folgendes Event hinzu:



Wenn wir in das Level mehrere Spawnpoints hinzufügen wollen, brauchen wir für diese Spawnpoints eine Objektvariable, die dem Player Objekt übergeben wird, sobald wir über den Spawnpoint gelaufen sind. Diese Objektvariable nennen wir „**spawn_point_id**“ und ändern die Instanzvariable der einzelnen Spawnpoints ab. Unser erster Spawnpoint hat dann die **ID 1**, der zweite die **ID 2** und so weiter.

Als Nächstes müssen wir noch die Information, bei welchen Spawnpoint unser Frosch als nächstes gespawnt wird, gespeichert werden. Hierfür geben wir unserem **PlayerObject** die Variable „**currentSpawnPoint**“. Damit dieser Wert auch überschrieben wird, legen wir hierfür wieder ein neues Event an, das wie folgt aussieht:



Jetzt könnt ihr theoretisch beliebig viele Spawnpoints in eurem Level setzen und so auch den Schwierigkeitsgrad entsprechend anpassen.

Wir wollen unser Level jetzt ein bisschen weiter ausschmücken, inzwischen haben wir schon neue Plattformen, wir haben Wasser wo unsere Spielfigur hineinfallen kann und Spawnpunkte.

Unser Level ist bis jetzt noch etwas klein angelegt, deswegen machen wir unser Level noch etwas größer. Wir fügen noch einen Hintergrund hinzu und um das Level noch lebendiger zu gestalten, fügen wir noch weitere Objekte und Assets als Dekoelemente hinzu.

Ihr könnt dabei euer Level nach Belieben gestalten und eurer Kreativität freien lauf lassen.

Schau auch mal im Asset Store von GDevelop vorbei, dort gibt es auch schon vorgefertigte Assets. Hier solltet ihr allerdings darauf achten, dass manche Assets Geld kosten und nicht kostenlos verfügbar sind, oder dass ihr die Ersteller der Assets angeben müsst.



Achtet auch darauf, dass ihr für die verschiedenen Elemente, die ihr in das Level einfügt, auch die richtigen Layer, also Ebene wählt, damit es im Spiel nicht zu Überlappungen kommt.

Wenn ihr euer Level so weit ausgeschmückt habt, wie es euch gefällt, lernen wir im nächsten Kapitel, wie wir mithilfe von einem Parallaxe Effekt, mehr Perspektive in die Szene bringen.



Ziele

In dieser Einheit solltet ihr gelernt haben wie man das „Plattform“ Behaviour setzt und wie man Spawnpoints in ein Level einbaut.



Aufgabe

Schmückt euer Level nach belieben weiter aus und fügt mindestens 2 verschiedene Spawnpoints hinzu, an denen die Spielfigur spawnnt sobald sie von einer Plattform ins wasser fällt.