



Einstieg

Im letzten Kapitel haben wir eingebaut, was passiert, wenn man ein Leben verliert. Jetzt wollen wir mal einbauen, was passiert wenn man Gewinnt. Wir wollen implementieren, dass man wenn man das erste Level schafft, direkt in das zweite Level kommt. Dafür wollen wir das erste Level mal duplizieren. Wir gehen zum **Project** Fenster unter *Scenes* mal zu unserem ersten Level. Dieses sollte Level1 heißen. Wenn es z.B. *SampleScene* heißt, nennen wir es zuerst in *Level1* um. Um dieses zu duplizieren, müssen wir Strg+d drücken. Dieses Level benennen wir mal zu *Level2* um. Wenn wir diese neue Szene öffnen, sehen wir es nun in der **Scene** Ansicht. Es sieht genauso aus wie Level 1, aber oben in der **Hierarchy** sehen wir das tatsächlich Level2 geöffnet ist. Damit wir die zwei Level voneinander unterscheiden können, entfernen wir im zweiten Level mal alle Plattformen. Wir gehen zuerst mal zu Level 1 zurück und erstellen dort am Ende des Levels eine neue Plattform aus einen der *Prefabs* die wir gespeichert haben. Um diese Plattform als Ziel zu markieren, setzen wir mal einen Stein drauf. Dieser befindet sich im *Art -> Decor* Ordner und heißt *stone*. Sobald wir diesen Stein erreichen, wollen wir ins zweite Level wechseln. Dafür erstellen wir ein neues Skript, welches wir im Ordner *Scripts -> Game* abspeichern und nennen es *LevelGoal*. Um das neue Skript zu schreiben, schauen wir uns mal das Skript vom Wasser nochmal an. Denn was wir beim Ziel machen wollen, ist eigentlich ähnlich zu dem, was auch beim Wasser passiert. Nur wollen wir, wenn der Spieler diesen Trigger berührt, nicht das Level neu starten sondern Level 2 starten. Dafür kopieren wir uns die Zeilen aus dem *water* Skript in das neue Skript.

```
Private void OnTriggerEnter2D(Collider2D collision)
{
    if (collision.tag == „Player“)
    {
        SceneManager.LoadScene(SceneManager.GetActiveScene().name);
    }
}
```

Das bauen wir jetzt so um, dass es für dieses Skript passt. Zunächst einmal haben wir hier den gleichen Fehler mit dem fehlenden Import vom *Scenemanager*. Das beheben wir auf die gleiche Weise wie zuvor, indem wir auf die Glühbirne drücken. Was wir nun machen könnten, wäre einfach den Namen des zweiten Levels in die *LoadScene()* Methode zu schreiben:



```
SceneManager.LoadScene(„Level2“);
```

Das Problem hierbei ist, dass wir das Skript vielleicht auch für zukünftige Levels benutzen wollen und dann ist es nervig es in ein anderes Level laden zu lassen als das zweite. Was wir aber machen können ist das Level in einer Variable zu speichern. Dafür erstellen wir eine neue Variable in der Klasse vom Typ String. Das bedeutet, dass wir hierin einen Text speichern können.

```
Public string levelName;
```

Jetzt können wir diese Variable in die *LoadScene()* Methode schreiben:

```
SceneManager.LoadScene(levelName);
```

Jetzt können wir von außerhalb einstellen in welches Level man kommen soll, sobald man das Ziel berührt. Das neue Skript packen wir nun auf unseren Stein und schreiben in das Feld *Level Name* den Text *Level2*. Dann fügen wir noch die *Box Collider 2D* Komponente hinzu und setzen dort ein Häkchen bei *Is Trigger*. Zuletzt nennen wir das Objekt noch zu *LevelGoal* um und erstellen ein *Prefab* davon. Damit es aber letztendlich funktioniert, müssen wir noch eine Sache machen. Nämlich alle *Szenes* die geladen werden sollen, müssen wir in den Build Settings hinzufügen. Dafür drücken wir oben links im Fenster auf *File -> Build Settings*. Jetzt ist für uns der obere Teil wichtig, der *Scenes in Build* heißt. Dort müssen nämlich alle *Szenen* rein, die wir benutzen wollen. In unserem Fall sind es *Level1* und *Level2*. Wenn dort noch was anderes drinnen ist, entfernen wir das zuerst mal. Um das zu machen, klicken wir die Szene in der Liste an und drücken die Entf-Taste. Um unsere Levels jetzt hinzuzufügen, müssen wir auf die *Add Open Scene* Taste drücken die sich unter der Liste befindet. Das Gleiche machen wir dann nochmal, während das zweite Level in der **Scene View** geöffnet ist. Wichtig ist, dass das erste Level oben auf der List ist, da das beim Start des Spieles geladen wird. Wenn wir das jetzt im Spiel ausprobieren, sehen wir wie wir in das zweite Level gelangen, sobald wir mit dem Frosch den Stein berühren. Mit diesen *Prefabs* als Bausteine können wir jetzt so viele Level bauen wie wir möchten und diese aneinander reihen.



Aufgabe

Füge ein Ziel am Ende des Levels hinzu. Wenn man das Ziel erreicht soll das zweite Level gestartet werden. Das Ziel soll im Spiel in der Form eines Steines markiert werden.