EINSTIEG

Bevor wir unser erstes Skript schreiben, brauchen wir zuerst einmal einen Ort, an dem wir es abspeichern können. Der Übersicht wegen erstellen wir dafür einen neuen Ordner mit dem Namen *Scripts*. In diesem Ordner erstellen wir einen weiteren Ordner mit dem Namen *Player*. Hier speichern wir alle Skripte ab, die mit dem Spielcharakter zu tun haben, wie zum Beispiel die Bewegung. Hier können wir jetzt ein Skript erstellen, indem wir nach dem Rechtsklick auf *Create -> C# Script* gehen. C# ist die Programmiersprache, in der man Skripte für Unity schreibt. Das Skript nennen wir *PlayerMovement*. Um das Script einem Objekt zuzuordnen, müssen wir es auf das Objekt in die Hierarchy ziehen. Verbinde nun das *PlayerMovement* Skript mit dem Player Objekt. Jetzt sollte im Inspector ein neuer Teil dazu gekommen sein, der dem Skript entspricht. Es ist auch möglich, ein Skript über den Inspector hinzuzufügen, indem man unten auf *Add Component* klickt und dort nach dem neuen Skript sucht.

Durch das Doppelklicken im Project Fenster kann man das Skript bearbeiten. Es öffnet sich eine Entwicklungsumgebung, in der man das Skript anpassen kann. Der für uns wichtige Teil der Entwicklungsumgebung ist der große weiße Kasten, der den meisten Platz einnimmt. Hier wird das Skript programmiert. Es steht bereits ein wenig Text in dem Fenster, der von Unity automatisch generiert wurde. Wenn wir uns den Text mal etwas genauer anschauen, sehen wir, dass die Klasse *PlayerMovement* Eigenschaften eines *MonoBehaviours* hat. Das ist grundlegend nötig, um das Script in Unity einbinden zu können. Auch wurden uns bereits zwei Funktionen generiert. Eine mit dem Namen *Start* und eine mit dem Namen *Update*. Die runden Klammern können wir zunächst mal leer lassen. Die werden später noch wichtig. Die geschweiften Klammern sind die, die für uns jetzt wichtig sind, denn da kommt rein, was die Funktion am Ende machen soll.

Die *Start()* Funktion wird einmal ganz am Anfang ausgeführt, wenn das Skript geladen wird. Die *Update()* Funktion wird einmal pro Frame ausgeführt. Spiele können verschieden viele Frames pro Sekunde haben. So haben sie typischerweise 30, 60 … Frames in einer Sekunde. Dementsprechend wird die Update Funktion dann 30, 60… mal in einer Sekunde ausgeführt. In die *Start()* Funktion können wir Dinge rein machen, die wir nur ganz am Anfang des Spiels brauchen, wie zum Beispiel, wo der Spieler erscheinen soll oder wie viele Leben er am Anfang hat. In die *Update()* Funktion können wir reinschreiben, was durchgehend gemacht werden soll. So soll immer überprüft werden, ob ein Knopf gedrückt wurde, damit wir darauf reagieren können und den Spieler dann z.B. dementsprechend bewegen können.

Nun fangen wir damit an, das mal zu implementieren. Wir wollen abfragen, ob gerade ein Button gedrückt wird. Dafür schreiben wir in die *Update()* Funktion die folgende Zeile Code;

```
Input.GetAxisRaw(„Horizontal“);
```

Hiermit fragen wir, ob gerade eine bestimmte Gruppe an Knöpfen gedrückt wird. In diesem Fall ist es der Pfeil-nach-links und der Pfeil-nach-rechts-Knopf, da diese für die horizontale Bewegung stehen. Um zu sehen, was wir da gerade durch die Zeile Code zurückbekommen, können wir diese mal in der Konsole ausgeben, indem wir das Folgende dazu schreiben:

```
Debug.Log();
```

Diese Funktion schreibt alles in die Konsole, das in die Klammern geschrieben wird. Wenn wir die Abfrage jetzt in die Klammern setzen, können wir sehen, was da rauskommt.

```
Debug.Log(Input.GetAxisRaw(„Horizontal“));
```

Wenn wir nun zurück in das Unity Fenster gehen und oben in der Mitte auf die Play Taste drücken, wird das Spiel gestartet und wir sehen, wie in der Konsole 0 ausgegeben wird, da gerade keine der Tasten gedrückt wird. Wenn wir jetzt die linke oder rechte Pfeiltaste drücken erscheint eine 1 oder -1 je nachdem welche Taste gedrückt wurde.

AUFGABE

Das Ziel dieses Kapitels ist es, die Benutzung von Skripten zu verstehen. Dazu gehören generelle Bestandteile und deren Funktionen. Dies dient zu Vorbereitung für die kommenden Kapitel.