



Einstieg

Nun kümmern wir uns um die Bewegung der Kamera. Damit sich die Kamera mit dem Spieler bewegt, ziehen wir sie in der **Hierarchy** mal auf den Spieler drauf. Dadurch wird sie ein Child-Objekt und wir sehen wie sie sich nun jedes Mal bewegt, wenn wir den Spieler bewegen. Damit der Frosch sich immer im Mittelpunkt befindet, müssen wir noch die **Transform** Komponente der Kamera anpassen. Dort setzen wir die x und y Position auf 0. Jetzt ist die Kamera immer direkt da, wo auch der Frosch ist. Wenn wir nun aber wollen, dass sich die Kamera nur auf der x-Achse mit dem Frosch bewegt, aber immer die gleiche Höhe halten soll, müssen wir es anders machen. Hierfür ziehen wir die Kamera wieder aus dem Player Objekt heraus. Dann erstellen wir unter dem *Scripts* Ordner einen neuen Ordner mit dem Namen *Camera*. Darin erstellen wir dann ein neues Skript, welches *CameraFollow* heißt. Das hängen wir dann an das *Main Camera* Objekt dran. In diesem neuen Skript erstellen wir zunächst einmal eine neue Variable, in die das Objekt kommen soll welches verfolgt wird:

Public Transform follow;

Dort ziehen wir im Editor dann den Player hinein, um eine Referenz auf ihn zu haben. Die Variable ist vom Typ Transform, weil wir auf eben diesen zugreifen wollen, um die Position des Spielers herauszufinden.

Wenn wir uns zurück an unser *PlayerMovement* Skript erinnern wissen wir, dass wir dort die Bewegung in der *FixedUpdate()* Funktion gemacht haben. Dies müssen wir hier genauso machen. In dieser Funktion erstellen wir aus dann wieder eine neue Variable vom Typ *Vector3*. Hier sind drei Werte drinnen gespeichert, einen x, y und z Wert. Das kommt uns doch bekannt vor. Genau die gleichen Werte sind auch im Transport für die Position gespeichert.

Void FixedUpdate()

```
{  
    Vector3 followPosition;  
}
```

Dort schreiben wir dann zunächst einmal die Position der Kamera hinein, indem wir die Variable wie folgend abändern:

Vector3 followPosition = transform.position;

Jetzt wollen wir den x Wert der Position ändern. Und zwar wollen wir da die x Position des Spielers drin haben.



```
followPosition.x = follow.position.x;
```

Nun haben wir einen neuen Wert, den wir wieder zurück in den Transform der Kamera schreiben wollen. Dies tun wir, indem wir das folgende schreiben:

```
Transform.position = followPosition;
```

Zusammengefasst kopiert das Skript jetzt immer den x Wert vom Spieler zu der Kamera. Das führt dazu, dass die Kamera sich immer auf der gleichen vertikalen Linie befinden wird wie der Spieler. Jetzt müssen wir nur noch die Höhe der Kamera so anpassen wie wir es wollen und dann schauen wir mal wie es im Spiel aussieht. Und wie wir sehen, bewegt sich die Kamera nun mit dem Spieler nach links und rechts. Die Höhe allerdings bleibt immer gleich. Wenn wir jetzt wollen würden, dass sich die Kamera nur nach oben und unten mit dem Spieler bewegt, müssten wir nur statt dem x ein y im Skript einsetzen und es würde funktionieren.

Eine weitere Einstellung die wir beeinflussen können, ist die Größe der Kamera. Die können wir in der *Camera* Komponenten auf dem *Main Camera* Objekt verändern. Wenn wir hier einen größeren Wert eingeben sehen wir, wie sich das Blickfeld der Kamera erweitert und viel mehr vom Level zu sehen ist. Einen guten Wert, den man hier wählen könnte, ist eine Größe von 7.

Der Hintergrund sieht jetzt aber noch recht leer aus. Wenn wir im zum Ordner *Art* -> *Background* gehen, finden wir eine Grafik mit den Namen *Matte*, die wir als Hintergrund einfügen können. Wenn wir sie in unser Spiel ziehen sehen wir, dass sie sich vor den anderen Grafiken befindet. Das wollen wir natürlich nicht. Wir wollen, dass sich das Hintergrundbild ganz hinten befindet. Um das zu machen, können wir im *Sprite Renderer* der Grafik *Additional Settings* aufklappen. Dort sehen wir ein *Order in Layer* Feld. Mit diesem können wir Unity sagen, welche Sprites hinter anderen verschwinden sollen. Eine größere Zahl heißt hierbei, dass es weiter vorne liegt. Für unsere Hintergrundgrafik geben wir dort mal einen Wert von -10 an.

Jetzt wollen wir noch, dass der Hintergrund die ganze Kamera ausfüllt, um das zu machen passen wir den *Scale* im *Transform* an. 1,5 ist hier ein guter Wert für den x und y Wert.

Beim Wasser können wir das Gleiche machen, wie beim Hintergrund. In diesem Fall wollen wir, dass sich das Wasser immer ganz vorne befindet. Hierfür geben wir beim *Order in Layer*, in der Wasser Grafik, den Wert 10 an. Zur Deko fügen wir jetzt auch noch ein paar Wolken hinzu. Diese befinden sich auch im *Background* Ordner und können von dort hineingezogen werden. Diese können wir so hinziehen, damit es uns gefällt. Dann ziehen wir die neuen Wolken Objekte auch



noch auf das Hintergrundobjekt in der Hierarchy, damit sie Child-Objekte werden. Damit sich die Wolken vor dem Hintergrund und hinter dem Wasser und den Plattformen befinden, stellen wir hier die *Order in Layer* auf -5. Auf die gleiche Weise kann man auch die Order der Wolken selber anpassen, wenn man da Änderungen haben möchte. Wenn wir uns jetzt jedoch im Spiel bewegen sehen wir, wie der Hintergrund plötzlich aufhört. Das wollen wir natürlich nicht. Wir wollen, dass sich der Hintergrund mit der Kamera bewegt. Wie das funktioniert, hatten wir schon mal. Denn um das zu erreichen, müssen wir einfach nur den Hintergrund auf die Kamera ziehen und ihn so zu einem Child Objekt machen.



Aufgabe

Stelle die Kamera so ein, dass sie sich horizontal mit dem Spieler bewegt damit wir später ein Level bauen können welches größer als nur ein Bildschirm ist. Füge dann einen Hintergrund mit Himmel und Wolken hinzu der sich auch mit der Kamera bewegt.