



2. Auflage

Spiele  
entwickeln  
mit

# UNREAL<sup>4</sup> ENGINE

jonas RICHARTZ

Programmierung mit Blueprints:  
Grundlagen & fortgeschrittene Techniken



Im Internet: Die Games zum Buch mit Projektdateien und weiterführende Videotutorials

HANSER

Ich werde in diesem Buch alle wichtigen Bereiche der Engine durchgehen: von einfachen Kapiteln im Bereich der Landschaftserstellung und Licht- und Schattengestaltung bis hin zu komplexeren Themen wie Animationen, Netzwerke und die erste einfache KI (künstliche Intelligenz). Du wirst eine komplette Übersicht zu allen relevanten Themen bekommen, dazu zwischendurch einigen Aufgaben mit Lösungsvorschlägen im Videoformat. Des Weiteren versuche ich, dir immer kleine Beispiele zu zeigen, womit du dir ein besseres Bild über die Funktionsweise und Nützlichkeit der einzelnen Aspekte machen kannst.

Wie du vielleicht schon festgestellt hast, schreibe ich dieses Buch sehr offen, als spräche ich mit dir persönlich. Es ist mir wichtig, dass du dich wohlfühlst und dass nicht alles in einem Fremdwortwirrwarr endet.

## ■ 1.3 Lizenzen

Wie bei fast jeder Engine üblich, musst du auch bei der Unreal Engine 4 gewissen Lizenzen zustimmen. Du kannst die Engine für nicht-kommerzielle und kommerzielle Projekte in den Bereichen der Spieleentwicklung kostenlos nutzen, aber auch für Film, Architektur und vieles mehr, was du mit der Engine anstellen kannst. Du darfst dir den vollen Source-Code kostenlos herunterladen und für den eigenen Nutzen abändern, jedoch nicht weiterverkaufen – es sei denn, du erstellst ein Plug-In, das du auf dem Unreal Engine Marketplace verkaufst. Es gibt hierbei keine verschiedenen Versionen der Engine, welche aufgeteilt ist in Free, Commercial oder Pro, sondern es gibt nur eine einzige Variante der Unreal Engine. Je nachdem, was du mit der Engine gestalten willst, ändert sich auch die Lizenz für dich, aber jedem Nutzer, sei es Privatperson, Schüler, Indie-Entwickler oder große AAA-Studios, steht die gleiche Engine zur Verfügung.

Solltest du dich dafür entscheiden, ein kommerzielles Produkt auf dem Markt zu bringen, das du mit der Unreal Engine 4 erstellt hast, musst du 5 % deiner Einnahmen an Epic Games abgeben, aber auch nur, wenn die Einnahmen in einem Quartal mehr als 3000 \$ betragen. Solltest du also 2500 \$ in einem Quartal verdienen, darfst du alles behalten. Solltest du 3100 \$ verdienen, musst du von den 3100 \$ an Epic Games 5 % abgeben. Genauere Informationen hierzu findest du auf Englisch unter [www.unrealengine.com/faq](http://www.unrealengine.com/faq).

Eventuell schreckt dich nun der Gedanke an die Abgaben ein wenig davon ab, die Unreal Engine 4 zu verwenden, aber bedenke, dass es sich hierbei um eine der besten Engines auf dem Markt handelt, mit der du die gleichen Optionen hast, ein Spiel zu entwickeln wie die großen Spieleentwickler, die ebenfalls die Unreal Engine 4 nutzen. Es gibt keinerlei Restriktionen, und du hast exakt das gleiche Tool wie alle anderen auch. Wenn du also das neue Unreal Tournament von Epic Games ansiehst, denk daran: Das kannst du auch.

Zusätzliche Inhalte, die ich im Beispielprojekt von Kapitel 22 erstellt habe, darfst du für dich nach Belieben in deinen eigenen Projekten nutzen, egal ob kommerziell oder nicht.

## ■ 1.4 Weiterentwicklung der Engine

Es ist üblich, dass sich Engines im Laufe der Zeit weiterentwickeln und neue Versionen erscheinen. Bei der Unreal Engine 4 kommt es alle ein bis drei Monate zu einer neuen Version, welche immer Verbesserungen und neue Features mit sich bringt. Als ich mit dem Buch anfang, war gerade einmal die Version 4.7 veröffentlicht worden, und mittlerweile steht Version 4.17 in den Startlöchern. Bis du diese Zeilen liest, hat sich eventuell sogar noch viel mehr geändert, und neue Features, von denen ich noch gar nichts wusste, sind erschienen, und ich hätte diese sicherlich gerne in diesem Buch beschrieben.

Du darfst dich also nicht wundern, wenn bei meinen Videos und Bildern nicht alles hundertprozentig so aussieht wie bei dir – das lässt sich leider nicht verhindern. In der Regel verändert sich jedoch das Aussehen nur geringfügig, und die jeweiligen Elemente sind oft nur einen kleinen Klick entfernt. Auch wenn ich nun schreibe, dass Feature X/Y noch nicht in der Engine oder in Blueprints verfügbar ist, kann es sein, dass es bei dir mittlerweile ganz anders aussieht und du dir denkst: Was hat der Typ denn für einen Unsinn geschrieben ... das ist doch alles hier!

Änderungen sind auf jeden Fall immer von Vorteil und helfen dir bei der Entwicklung, aber es kann auch hin und wieder vorkommen, dass mit einer neuen Engine Version einige deiner Blueprints aufhören zu funktionieren. Dies geschieht in den seltensten aller Fälle, und ich selbst habe es bisher auch nur ein oder zweimal erlebt, aber es kann dennoch passieren. Dann musst du genau nachschauen, an welcher Stelle das Blueprint aufhört zu funktionieren, und dein Blueprint dahingehend anpassen. Dabei kann es sich auch um etwas ganz Banales handeln, wenn z.B. die Verknüpfung zweier Funktionen gekappt wurde. Aber manchmal erfordert das Beheben solcher Probleme ein wenig mehr Vorwissen. Für den Anfang rate ich dir deswegen auch, auf einer Engine-Version zu bleiben, bis du dich mit deinen Fähigkeiten wohl genug fühlst, ehe du dir die neueste Version herunterlädst. Wenn du aus diesem Buch lernst, welches hauptsächlich mit der Version 4.8 – 4.9 erstellt wurde, wirst du dich auch in den Versionen am besten zurechtfinden können. Wie bereits erwähnt, kann ich nicht garantieren, dass sich zukünftige Versionen optisch und funktionell nicht verändern. Mit der zweiten Auflage habe ich jedoch versucht alle Änderungen bis 4.16 zu berücksichtigen und Bilder gegebenenfalls zu aktualisieren.

Aber ich will dir jetzt auch nicht zu viel Angst machen, dass du gar nicht mehr mit der Entwicklung mithalten wirst, wenn du zum ersten Mal damit arbeitest. Aber wenn du etwas nicht auf Anhieb findest, haben sich möglicherweise nur der Ort oder die Bezeichnung ein wenig verändert.

Für alle Fälle werde ich auf meinem YouTube-Kanal immer wieder Updates veröffentlichen und neue Videos herausbringen, um dich auf Änderungen aufmerksam zu machen und dich so weiterhin bei der Entwicklung deines Spiels zu begleiten.

# 2

## Erste Schritte

Ich werde nicht nur über die Entwicklung mit der Unreal Engine 4 reden, sondern dir auch ein bisschen von der Philosophie bei der Spieleentwicklung ans Herz legen. Es gibt extrem viel zu beachten, und das alleinige Wissen über Mechaniken wird dich nicht überall hinbringen und garantiert kein erfolgreiches und spaßiges Spiel. Klar muss ich dir jetzt nicht erklären, dass es nicht so toll ist, wenn dein Spiel voller Fehler ist und dein Charakter andauernd durch den Boden fällt. Es geht mir vielmehr darum, dass du immer genau weißt, warum du etwas Bestimmtes machst und welchen Nutzen du und der Spieler daraus erhalten. Das Wichtigste ist, dass du Spaß an der Entwicklung hast und der Gamer Spaß am Spielen, dann sind alle Beteiligten glücklich und somit hoffentlich erfolgreich.

### ■ 2.1 Wie fange ich an?

Es kommt immer darauf an, welches Spiel du machen willst und welche Erfahrung du mitbringst. Der logische Anfang wäre natürlich, dieses Buch zu lesen, denn einfach ohne jedwedes Vorwissen draufloszulegen, wird dir nur Probleme bereiten.

Es lohnt sich nicht, gleich von Anfang an dein Spiel der Träume zu bauen. Setze dir vielmehr zuerst kleine Ziele. Wenn du gerade das erste Mal mit der Engine anfängst und denkst: „Geil, dann mach ich jetzt mein Ultra-MMO mit zerstörbarer Landschaft und einem Minimum von 1000 Stunden Spielzeit“, dann hast du dir ein wenig zu viel vorgenommen. Fang immer klein an, vor allem wenn du noch am Lernen bist. Stelle dir kleine Aufgaben, die du nacheinander erfüllst. Beispielsweise könnten deine ersten Aufgaben darin bestehen, kleine Features wie das Öffnen und Schließen einer Tür oder das Ein- und Ausschalten mehrerer Lichter in einem Raum mit einem Lichtschalter einzubauen. Wie genau du sowas angeht, wirst du im Laufe dieses Buches noch lernen, aber das meine ich mit „klein anfangen“. Wenn du dann irgendwann soweit bist und alle Grundlagen verstanden hast, kannst du anfangen, kleinere Spiele zu bauen.

Neben der Idee brauchst du noch einen gewissen Plan, wie du die Aufgabe angehen willst. Einfach drauflos zu bauen, kann funktionieren, ist aber nicht empfehlenswert. Du musst dir genaue Gedanken machen, wie du deine Aufgabe angehen willst und was du dafür machen

musst. Bleiben wir bei dem simplen Beispiel, einen Lichtschalter für Räume einzubauen. Du benötigst ein Lichtschalter-Objekt, Lichter und die Logik, um diese Lichter ein- und ausschalten zu können. Für die Funktion brauchst du eine Möglichkeit, dass dein Charakter mit dem Lichtschalter kommunizieren kann. Der Lichtschalter muss mit den jeweiligen Lichtern kommunizieren können und wissen, ob er die Lichter an- oder ausschalten soll usw.

Liegen all diese Fragen auf dem Tisch, liegt es an Dir, das Puzzle zusammenzufügen und herauszufinden, wie du die einzelnen Elemente am besten verbinden kannst. Dies kann vor allem bei komplexen Aufgaben und Funktionen helfen.

## ■ 2.2 Motivation

Motivation ist ein wichtiger Faktor bei der Spieleentwicklung. Ich habe schon oft erlebt, wie mangelnde Motivation ein Projekt zum Scheitern gebracht hat. Es gibt ein paar Tricks, wie du deine Motivation aufrechterhalten kannst.

Zum einen ist Abwechslung ein guter Faktor. Wenn du mehrere Wochen an einem neuen und innovativen Feature sitzt, es aber einfach nicht so funktionieren will, wie du gerne hättest, dann lass es erstmal. Such dir eine andere Aufgabe, bring dich auf andere Gedanken und mach deinen Kopf frei. Wenn man sich zu sehr auf etwas versteift, gibt es nur Probleme. Kommst du nach einer kleinen Auszeit wieder an diese Aufgabe zurück, hast du eine vollkommen andere Sicht auf das Geschehen. Der Tunnelblick ist verschwunden, und du hast neue Ideen, dein Problem zu lösen.

Finde Teammitglieder, die genauso motiviert sind wie du! Alleine zu arbeiten kann auch Unmengen an Spaß machen – vor allem, weil man sein eigener Herr ist, aber auf Dauer ist es doch schön, andere Teammitglieder zu haben, als alles alleine zu erledigen. Aber auch hier musst du vorsichtig sein, denn Teammitglieder können auch eine große Motivationsbremse sein. Wenn Aufgaben nicht erledigt werden oder an dir hängen bleiben, keine Kommunikation herrscht und jeder macht, was er will, dann ist das mehr als nervig. Versuche also, deine Teammitglieder gut auszuwählen, und nicht nur nach deren Talent, sondern eher anhand ihrer Motivation. Jeder hat die Möglichkeiten, sein Talent zu verbessern, aber nicht jeder kann seine Motivation für längere Zeit halten.

Nimm dir nicht zu viel vor! Wenn du dir selbst direkt einen Berg an Aufgaben gibst, kannst du schnell die Übersicht und Motivation verlieren. Kleine Schritte führen zum Erfolg!

## ■ 2.3 Planung

Wenn du in einem Team arbeitest, ist es wichtig, dass du und deine Teammitglieder einen Plan haben. Nicht nur, was für Aufgaben momentan erledigt werden sollen, sondern auch, welche Aufgaben schon erledigt sind und wer welche Aufgabe zugewiesen hat. Somit hast

du immer einen genauen Überblick über den momentanen Entwicklungsstand und kannst sehen, an welcher Stelle es hakt und wo es gut läuft. Dafür steht dir auch eine große Bandbreite an Hilfsmitteln zur Verfügung, um deine Planung übersichtlich darzustellen.

Zum einen gibt es das kostenlose Web-Tool namens **Trello**. Auf **Trello** können neue Karten in Kategorien erzeugt und bei Beendigung in eine andere Kategorie verschoben werden. Von der Kategorie 3D-Models kann die Aufgabe Charakter erstellen in die Kategorie Erledigt bugsiert werden. Wie genau deine Kategorien heißen und wie du diese managst, bleibt dir natürlich komplett offen.

Für eine professionellere, nicht kostenlose, aber relativ günstige Möglichkeit gibt es noch **Unfuddle**. Ich persönlich benutze **Unfuddle** ebenfalls. Hier hat man die Möglichkeit, übersichtliche Tickets zu erstellen und diese Personen im Team zuteilen zu können. Sobald ein Teammitglied einen Bug findet, kann dieser dazu ein neues Ticket erstellen und es einer Person zuweisen. Dann erhalte ich auch direkt eine Mail mit dem Ticket und kann sehen, dass es einen Fehler beim Inventarsystem gibt, durch den man Gegenstände nicht mehr ablegen kann. Dann kann ich das Ticket akzeptieren, das Problem lösen und auf z.B. Testing stellen. Die anderen Mitglieder können das natürlich auch sehen, testen und bestätigen, dass der Fehler erfolgreich behoben wurde. Dann wird das Ticket archiviert, und man macht wie gewohnt weiter. Das nur als kleines Beispiel wie man die ganze Geschichte handhaben kann. Zusätzlich kann man dort noch Milestones einstellen (Zeitangaben, wann bestimmte Features fertig sein sollen), damit man ein Ziel hat, auf das hingearbeitet wird.

Es gibt natürlich unzählige verschiedene Seiten und Programme, die du benutzen kannst, und es ist egal, ob du das mit einem tollen Programm erstellst oder alles in einer eigenen Excel-Tabelle verwaltest. Die Hauptsache ist jedoch, dass du einen Überblick und richtige Planung über die Entwicklung hast.

## ■ 2.4 Sicherheitskopien

Es kann immer zu Problemen kommen, die dein komplettes Projekt lahmlegen können. Sei es, dass du dein eigenes Projekt nach einem Absturz nicht mehr öffnen kannst, oder deine Festplatte den Geist aufgibt: Möglichkeiten, dein Projekt zu zerschließen, gibt es viele. Deswegen ist es immens wichtig, dass du Sicherheitskopien anlegst – am besten auf einen separaten Server.

Mithilfe von Tools wie **SVN** oder **Git** kannst du dein Projekt gesichert auf deinem eigenen Server lagern und anderen Personen Zugriff auf eben dieses Projekt geben. Jeder kann sich somit das Projekt herunterladen, Änderungen durchführen und diese auf den Server hochladen. Sollte also dein PC in die Luft fliegen, kannst du unbesorgt sein (außer natürlich du sitzt mitten in der Explosion), da deine Daten sicher sind und nichts verloren gegangen ist. Sollte es aber auch passieren, dass du aus Versehen etwas hochlädst, was dein Projekt unbrauchbar machst, hast du mit **SVN** und **Git** die tolle Möglichkeit, ein paar Versionen zurückzugehen, oder nur bestimmte Dateien auf einen früheren Stand zurückzusetzen.

Im optimalen Fall musst du natürlich nicht darauf zurückgreifen, und alles läuft von Anfang an reibungslos, aber es ist immer sehr hilfreich zu wissen, dass deine Daten sicher sind.