

Screenshot einer 3D Szene mit verschiedenen Audioquellen (Lautsprechersymbole), deren Triggerzonen (grüne Boxen) und Distanzradien (hellblaue Kugeln)

Kreative Tongestaltung in der Game Audio Produktion

Interaktive Medien wie Videospiele, ob auf dem Bildschirm, in Virtual Reality oder in Augmented Reality, bieten Sound Designern spannende Möglichkeiten und Herausforderungen für die kreative Tongestaltung. Dieser Artikel beschäftigt sich mit den grundlegenden Unterschieden zum klassischen, linearen Film und erklärt die Konzepte und Herangehensweisen adaptiver Klangkulissen.

Text David Schornsheim

Grundlagen Film versus Videospiegel

Videospiele funktionieren gänzlich anders als Filme: Filme sind linear, Videospiele interaktiv. Das heißt, dass man auf den Verlauf von Filmen als Zuschauer keinerlei Einfluss nehmen kann. Sämtliche Elemente eines Films werden in Bezug auf Timing und Ablauf bei jedem Durchlauf unverändert wiedergegeben. Zwar gibt es auch Formate, in denen Zuschauer durch Entscheidungen den Hauch einer Mitbestimmung der Geschichte spüren (zum Beispiel *Black Mirror: Bandersnatch*, Netflix), doch diese wählen lediglich eine von zwei nachfolgenden, vorproduzierten Szenen aus. Das ist vergleichbar mit interaktiven Büchern, bei denen der Leser vor einer Entscheidung

steht und dann entweder auf Seite x oder auf Seite y weiterlesen soll. Die Texte, die dann die Geschichte fortsetzen, sind bereits fertig geschrieben und gedruckt.

Bei Videospiele ist das prinzipiell anders, denn hier werden permanent Eingabebefehle an das Spiel gesendet, welches darauf reagiert. So entsteht eine direkte Verbindung zwischen Spiel und Spielenden. Die Aktionen der Spieler haben einen direkten Einfluss auf z. B. den Charakter auf dem Bildschirm, den zeitlichen Fortschritt des Spiels oder den Verlauf der Geschichte. Daher bezeichnen wir Videospiele als interaktiv.

Die Arbeit an Filmen ist zeit- bzw. frame-bezogen. Die Einzelbilder eines Films sind

anhand der Framerate in Stunde, Minute usw. eindeutig zugeordnet. Wird bei Frame 00:30:25:13 eine Tür geschlossen, beginnt an dieser Stelle im Ton in der Regel ein entsprechendes Geräusch.

Videospiele hingegen funktionieren nicht so, auch wenn es hier ebenfalls Frames gibt. Ein Spiel ist eine große Ansammlung von Regeln in Form von Programmcodes und Assets wie 3D-Modellen, Bildern oder Klängen. Etwa sechzig Mal pro Sekunde werden diese Regeln von den Prozessoren ausgewertet, auf dessen Grundlage ein Bild berechnet und der Audiobuffer gefüllt. Relevant für die Gestaltung bei Spielen sind daher Events und Parameter, letztere oft in Form von States. Da man aufgrund der



Abb. 1: Screenshot aus *Red Dead Redemption 2*

interaktiven Natur nie vorhersagen kann, bei welchem Frame eine Tür zugehen wird, löst man stattdessen in diesem Moment in einem System ein Ereignis aus, über welches das Audiosystem informiert wird und beginnt, das Türgeräusch abzuspielen. Wurde die Tür langsam geschlossen, drückt ein Parameter mit niedrigem Zahlenwert aus, ein sanftes Schließgeräusch wiederzugeben. Ist die Tür danach zu, kann auch ein State geändert werden und das Audiosystem lässt die Geräusche außerhalb der Tür gedämpft klingen.

Diese Unterschiede beeinflussen dementsprechend die Entwicklung des jeweiligen Mediums. Während sich beim Film der klassische Ablauf grob in aufeinander aufbauende Meilensteine herunterbrechen lässt (zum Beispiel: Idee -> Drehbuch -> Dreh -> Schnitt -> Ton und Musik), sind die Gewerke bei der Spielentwicklung deutlich enger verzahnt und unterliegen einem stetigen Wandel. Spielentwicklung ist Softwareentwicklung – die Arbeit im Tonbereich für Videospiele unterscheidet sich daher enorm von der Arbeit am Film. Videospiele werden mit sogenannten Engines erstellt. Dies sind Programme, welche die Regeln des Spiels und die Assets wie 3D-Modelle oder Sounds verwalten, in Szenen editierbar machen und zu einem fertigen Programm zusammenbauen. Der Großteil an Spielen wird mit Unity oder Unreal entwickelt. Einige wenige, meistens sehr große Studios, können es sich leisten, eigene Engines speziell für die Bedürfnisse der Projekte zu programmieren. Sowohl Unity als auch Unreal

besitzen vielfältige Möglichkeiten für die Soundgestaltung.

Es gibt aber auch dedizierte Audiosysteme wie Wwise oder FMOD, die sich in Engines integrieren lassen. Alle genannten Lösungen haben sich über Jahrzehnte in der Praxis bewährt, sind ohne Kosten installierbar und gut dokumentiert. Um ein Gefühl dafür zu kriegen, welches Tool für welchen Anwendungszweck geeignet ist, kann man sich auch ansehen, welche Videospiele damit entwickelt wurden. Prinzipiell lassen sich aber alle grundlegenden Audio-Workflows für Videospiele mit all diesen Tools umsetzen.

Game Audio

Game Audio ist Sound Design für interaktive Medien und ist im Grunde eine Event- und parameterbasierte Art der Tongestaltung in Echtzeit. Ein Event ist nur ein Auslöser – „etwas ist passiert“. Sobald das Audiosystem ein Event vom Spiel empfängt, muss es entsprechend reagieren. Parameter steuern dabei Details wie die Lautstärke. Ein Parameterwert von 0 könnte ein Geräusch unhörbar machen, weil es zu weit weg ist, während ein Parameterwert von 1 es maximal laut macht, weil es nah oder wichtig ist. All diese Regeln wie: Auf welche Events muss das Audiosystem reagieren, mit welchen Klängen muss es reagieren und wie sollen diese Klänge auf weitere Parameter des Spiels reagieren, werden vom Sound-Team festgelegt und sind der grundlegende Teil interaktiver Tongestaltung.

Video 1: *Red Dead Redemption 2*:

Information Cues: Ein nicht in der Spielwelt verorteter Percussion-Sound lenkt die Aufmerksamkeit auf einen Hinweis im User Interface. Die einzelnen Klangelemente der Atmosphäre werden zur Laufzeit nach Regeln vom Spiel ausgewählt und zusammengemischt. Dabei spielen Parameter wie die Distanz zu einem sprechenden Charakter eine Rolle.



<https://vimeo.com/694444357/c9e0561441>

In der Szene aus dem Action-Adventure *Red Dead Redemption 2* von Rockstar Games (siehe Video 1) reitet der Charakter nachts durch eine Schneelandschaft. Zu hören sind Windsturm, Wölfe und eine leicht beunruhigende Musik. Der Klang der Atmosphäre wird durch Regeln definiert. Dazu gehören zum Beispiel Uhrzeit, Ort, Witterung oder Fortschritt in der Story. Derselbe Ort kann im Spiel zu einer anderen Tageszeit besucht werden und hört sich dann entsprechend anders an.

Zusätzlich werden Foleys nach anderen Regeln und Synchronisationspunkten abgespielt. Im Beispiel bestehen diese aus den Fragen: Wann genau berührt ein Huf des Pferdes den Boden? Woraus besteht der Boden? Wie tief ist der Schnee? Da wir in einer Winterlandschaft sind, wird das Soundset für Schnee aktiviert. Später im Spiel reiten wir durch sommerliche Gras- oder Felslandschaften. Dann werden andere Soundsets verwendet. In der Szene ist der Charakter auch nicht alleine. Neben ihm reiten weitere Männer. Hier kommt noch ein Parameter für die Distanzabhängigkeit dazu. Je nachdem wie weit der Spieler von den anderen Charakteren entfernt ist, desto deutlicher oder undeutlicher ist deren Stimme hörbar. Eine weite Distanz kann auch dazu führen, dass der Dialog gar nicht abgespielt wird. Erst wenn sich der Spieler wieder nah genug an den anderen Charakteren befindet, würden die Gespräche begonnen oder fortgesetzt.



Abb. 2: Screenshot aus *Grand Theft Auto V*

In Spielen werden außerdem oft Hinweise und Tipps außerhalb der Spielwelt eingebunden, so als würde man sie auf den Bildschirm geklebt haben. Dies wird als User Interface bezeichnet. Die Hinweise bestehen zum Beispiel aus einer Erklärung, welche Taste gedrückt werden muss, um eine Aktion durchzuführen wie *Drücke W, um Dutch zu folgen* (siehe Abb. 1). Dutch ist in diesem Fall ein nicht spielbarer Charakter. Für diese Hinweise wird ein Percussion-Sound eingesetzt, um die Aufmerksamkeit der Spieler darauf zu lenken. Die Möglichkeit, mit dem richtigen Sound auch anderweitig Informationen über die Welt zugänglich zu machen, wird in Videospielen routiniert und kreativ genutzt und macht Sound zu einem extrem wichtigen Bestandteil interaktiver Erfahrungen.

Audio Listener und Audio Source

Eine zentrale Arbeit in der Tongestaltung ist die der Perspektive. Im Film werden die Zuschauer durch die O-Ton-Aufnahmen, Sound Design sowie Mischung in eine nachträglich unveränderbare, filmdienliche Hörperspektive versetzt.

Bei Videospielen wäre eine nachträglich unveränderbare Perspektive äußerst unpassend, denn wenn sich Spieler während eines Dialoges lieber in den benachbarten Raum begeben, um in Regalen nach Dingen zum Einsammeln zu suchen, sollte sich die Stimme des Dialogpartners vielleicht aus dem off anhören und auch die Position anpassen, statt mittig und präsent zu bleiben wie ursprünglich geplant. Daher gibt es hier das Konzept der Audio Source, also der Geräuschquelle, und des Audio Listeners, was man sich wie ein Mikrofonset in der virtuellen Welt vorstellen kann, welches

die unterschiedlichen Geräuschquellen aufnimmt.

Anhand der Beziehung zwischen Audio Source und Audio Listener werden für jede einzelne Quelle die Parameter der Mischung und räumlichen Effekte bestimmt. Dazu gehören die Richtung der Schallquelle, die Verdeckung unter Einbeziehung des Physiksystems oder auch die Distanz und vieles mehr.

Eine wichtige Entscheidung hierbei ist, wo der Audio Listener platziert wird. Im Regelfall ist die Platzierung auf der Kamera sinnvoll, je nach Genre und Intention muss die Antwort aber nicht so einfach ausfallen. In einem Spiel aus der Ich-Ansicht, der sogenannten First-Person, ist der Audio Listener auf der Kamera naheliegend und üblich, da die Spieler die Welt direkt aus den Augen der Spielfigur betrachten. Bei einem zweidimensionalen Sidescroller wie *Super Mario Bros.*, wo sich die Kamera nur nach links oder rechts bewegt, würde diese Entscheidung ebenfalls sinnvoll sein. Allerdings wird hier in der Praxis häufig auf räumliche Bezüge wie Verdeckung oder Distanz verzichtet. Auch bei einem Spiel in der Ansicht aus der dritten Person, der sogenannten Third-Person, bei der die Kamera der Spielfigur hinterher fährt, würde man den Audio Listener wohl auf die Kamera setzen. Sonst würde ein Geräusch, das klar auf dem Bildschirm, also vor den Spielern, zu sehen ist, auf einmal von hinten zu hören sein, nur weil sich der Charakter mit dem Rücken zur Geräuschquelle dreht. Also muss der Audio Listener auch hier auf die Kamera.

Video 2: *Grand Theft Auto V*.

Listener Parameters: Die Richtungslokalisierung des Springbrunnens wird von der Kamera bestimmt, die distanzabhängige Lautstärke hingegen von der Position des Charakters.



<https://vimeo.com/694435304/9683315227>

In dem Beispiel aus *Grand Theft Auto V* (siehe Video 2) kommt der Klang des Springbrunnens immer aus der Richtung, aus der er im Bild zu sehen ist. Die Richtung bezieht sich also auf die Kamera. Aber der Klang wird nicht lauter, wenn die Kamera dem Springbrunnen näher kommt, sondern der Charakter. Der Audio Listener für die Distanz befindet sich also auf dem Charakter und nicht auf der Kamera. Diese Entscheidung ist ein äußerst gelungener Kompromiss: Mit der Maus kann eine Kamera recht schnell bewegt werden, was zu rabiaten Lautstärkeänderungen führen würde und den Mix sehr unruhig wirken ließe, da die Bewegungen der Kamera zu stark betont wären. Es kann also sinnvoll sein, die einzelnen Mischungparameter von unterschiedlichen Objekten abhängig zu machen.

Leider unterstützen viele Game und Audio Engines einen derartigen Workflow mit mehreren Audio Listener Parametern nicht. Oft ist es für Third-Person-Spiele aber schon ausreichend, den Listener zwischen die Kamera und den Charakter zu platzieren.

Parameterautomatisierung

Unter dem Begriff Parameterautomatisierung versteht man beim Film, die relevanten Mischungseinstellungen über den zeitlichen Verlauf aufzuzeichnen und zu fixieren. Das Resultat wird letztendlich als eine statische Audiodatei herausgerechnet. Aufgrund der Interaktivität müssen bei Videospielen diese Mischeinstellungen als Regeln festgelegt und zur Laufzeit immer wieder neu angewandt werden. So wird die Mischung eines Videospieles über die periodisch ausgewerteten Parameter automatisiert. Allerdings ist die Mischung nur

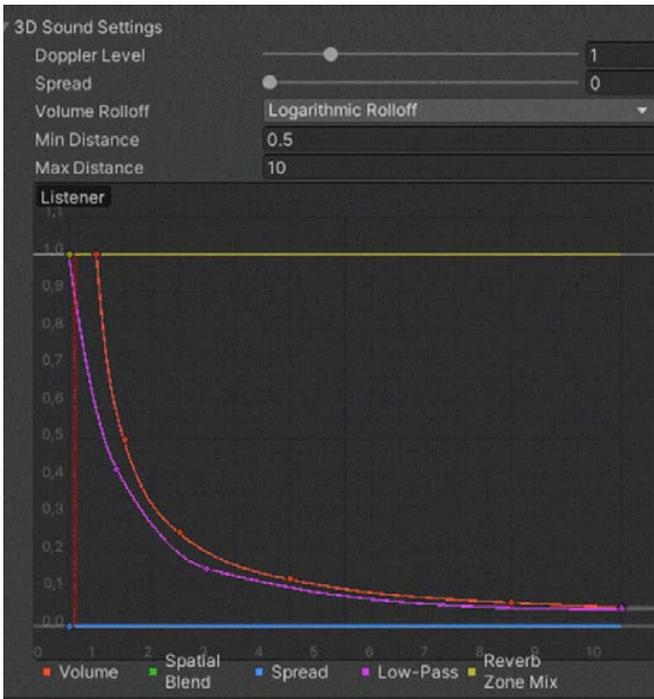


Abb. 3: Distanzparameter einer Audio Source der Engine Unity

ein Beispiel, tatsächlich gilt dies auch für grundlegendere Dinge, beispielsweise die Kopplung der Tonhöhe eines Motorengeräusches an die Fahrgeschwindigkeit eines Autos statt wie beim Film an die aktuelle Zeit- bzw. Framenummer.

Welche Parameter man für die Tongestaltung eines Spiels auswählt, ist Teil des kreativen Prozesses und muss für jedes Spiel neu entdeckt werden. Beispielsweise wären anhand der Distanz zu einer Schallquelle die in Abbildung 3 dargestellten Parameter möglich.

In Abbildung 3 sind die Distanzparameter einer Audio Source der Engine Unity zu sehen. Die folgenden Erklärungen dienen nur als Beispiel, sind teilweise vereinfacht und keine Aufforderung dazu, jede Quelle mit all diesen Abhängigkeiten einstellen zu müssen. Immer zweckdienlich denken! Die rote Kurve gibt die abfallende Lautstärke mit zunehmender Entfernung an. Interessant hierbei ist, dass die Maximallautstärke schon bei einem Meter erreicht ist. Oft genügt es, keine weitere Lautstärkezunahme im Zentimeterbereich um den Kopf darzustellen. Dies lässt sich auch mit der Einstellung *Min Distance*, also der minimalen Distanz, beeinflussen – alles, was kleiner als die *Min Distance* ist, wird nicht mehr geändert. Genauso verhält es sich mit der *Max Distance* (maximale Distanz), nur andersrum.

Mit der blauen Linie *Spread* lässt sich die Ausdehnung eines Geräusches in der Breite darstellen. Ist das Geräusch weiter

weg, könnte man die, vereinfacht gesagt, „Stereo-Breite“ einengen. Das passiert im Beispielbild aber nicht. Das Geräusch ist immer eine Punktschallquelle.

Die lila Kurve gibt den Frequenzverlauf eines Low-Pass Filters an und ahmt die mit steigender Entfernung zunehmende Luftschalldämpfung nach. Wichtige Geräusche oder Dialoge müssen vielleicht nicht ganz so starke rote und lila Kurven haben wie Hintergrundgeräusche.

In Verbindung mit einem Mehrkanal-Setup oder einer binauralen Mischung entsteht so ein dreidimensionaler Informationsraum um die Spieler.

Bei all diesen tollen Möglichkeiten sei dennoch darauf hingewiesen, dass Realismus ebenso wie beim Film in der Regel nicht die absolute Wahrheit in der Tongestaltung ist. Es ist nur gut, was gut klingt, bzw. was passt.

Die anderen Kurven lassen wir jetzt mal beiseite – es geht hier ums Prinzip. Und nicht vergessen, hier ging es nur um die klanglichen Auswirkungen durch den Distanzparameter. Man kann und sollte Geräusche von verschiedenen Parametern eines Spiels abhängig machen.

In Abbildung 4 sieht man, wie mehrere Geräuschquellen, hier dargestellt mit Kugeln, in einer Szene platziert sind. Jede Quelle hat eigene Distanzparameter, erkennbar an den unterschiedlichen Kugelgrößen. Die zwei grünen Boxen stellen Triggerbereiche dar, in denen die Sounds aktiviert werden.

Der Screenshot aus dem Unity-Editor zeigt das kurze Spiel *Sacrifice* (<https://pepper-mint-audio.itch.io/sacrifices>), bei dem die Spieler mit jeder Entscheidung einen Einfluss auf den Klang und die Zusammensetzung der Geräuschkulisse nehmen, die aus über 130 Audio Sources besteht.

Formate

Die verwendeten Formate sind in Videospielen nicht groß anders als beim Film. Ausgangsseitig sind die meisten Game und Audio Engines flexibel und können von Stereo über 7.1 und teilweise bis hin zu Atmos alles bedienen. Die benötigten Informationen für solche Mixes befinden sich ja bereits im 3D-Raum der Engine.

Als Eingangsformate während der Produktion wird vor allem Mono genutzt, da es von der gut lokalisierbaren Punktschallquelle bis zu einer umhüllenden Punktschallquellenwolke einsetzbar ist. Aber auch Mehrkanalformate wie 7.1 kommen für umhüllende Ambiances in Frage. Stereo kann sowohl für Musik als auch simple „breite“ Schallquellen zum Einsatz kommen.

Besonders hervorzuheben ist noch Ambisonics, welches von allen Mehrkanalformaten das flexibelste ist, da es von Mono, über binaural bis zu Atmos auf alle Systeme skaliert.

Seltener gibt es rein synthetische Geräuschquellen, sogenannte prozedurale Sounds. Diese werden mit Programmcode oder in grafischen, modular-artigen Synthesizerumgebungen entwickelt. Früher, als freier Speicherplatz noch im Kilo- oder Megabytebereich gezählt wurde und daher kein Platz für Audiodateien vorhanden war, waren solche Klänge sehr viel üblicher als heute.

Workflows

Das menschliche Gehör kann unveränderte Wiederholungen von kurz zuvor gehörten Geräuschen sehr schnell erkennen. Das wird auch als *Machine-Gun-Effekt* bezeichnet. Da man in Videospielen Aktionen oft relativ häufig hintereinander durchführt, ist es naheliegend, hierbei etwaigen Ermüdungserscheinungen des Gehörs vorzubeugen. Dafür kann man mehrere Variationen eines Geräusches anfertigen und zur Laufzeit nach Zufallsregeln entscheiden, welches der Geräusche abgespielt wird. Zusätzliche Randomisierung von Parametern wie der Tonhöhe können bereits aus einem oder wenigen Geräuschen größere

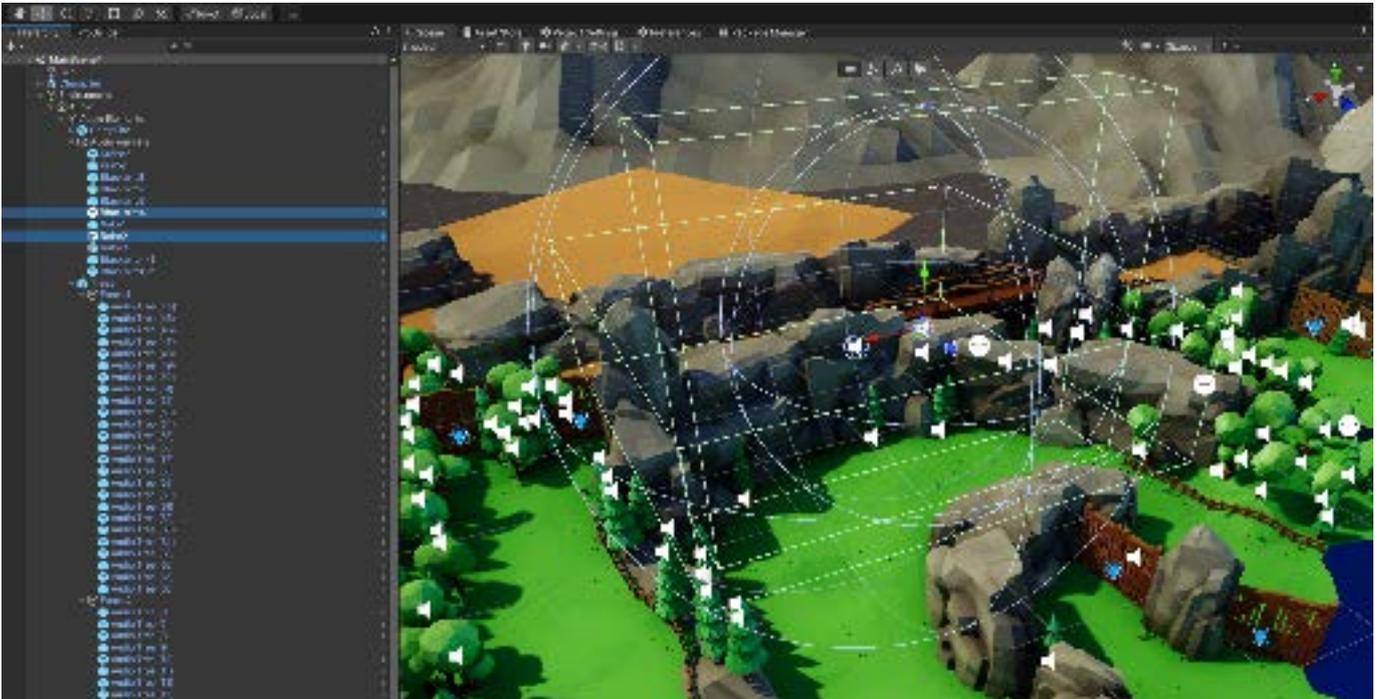


Abb. 4: Mehrere Geräuschquellen, hier dargestellt mit Kugeln, sind in einer Szene platziert

Variationsmöglichkeiten erzeugen. Selbst ein so einfacher Parameter wie die Abspieldauer kann dabei hilfreich sein.

Der Sprung-Sound aus *Super Mario Bros.* (1985), der an eine Kolbenflöte erinnert, ist pro Minute mehrmals zu hören, denn in diesem Spiel macht man eigentlich nichts anderes als zu springen. Allerdings wird der Sprungsound nur so lange gespielt, wie Mario an Höhe gewinnt. Ein kurzer Sprung klingt daher leicht anders als ein langer Sprung. Auch ist der Klangcharakter und die Tonhöhe des Sprungsounds abhängig von der Größe Marios, was zusätzliche klangliche Variation erzeugt.

Das separate Ausspielen von Layern und Steuern der einzelnen Layer durch Parameter bietet sich für adaptive Klangkulissen geradezu an. So kann ein Aufschlag auf dem Boden durch Verstärken eines separaten Layers stufenlos nasser oder trockener klingen oder eine Maschine immer komplexer, je nach gedrücktem Schalter. Auch Loops sind ein wichtiger Bestandteil interaktiver Tongestaltung und eignen sich vor allem für Elemente einer Umgebungsatmosphäre oder das Bestehen eines Zustandes wie das Einschalten der Stromversorgung.

Dabei sollte man immer einen Blick auf die Hardware-Ressourcen der Zielplattform haben, sonst kann der Prozessor, der sich neben dem Sound auch um Eingabe, Grafik, Physik usw. kümmern muss, die Aufgaben nicht schnell genug abarbeiten und das Spiel stockt.

Diese Konzepte sind nicht grundlegend neu und werden auch im Filmbereich verwendet. Die Herausforderung bei Spielen liegt aber darin, diese interaktiv zu machen. Diese Techniken müssen so angewandt werden, dass die Vision der zum Spiel passenden Klangkulisse mittels Events und Parametern ausgedrückt werden kann.

VR und AR

Auch wenn Virtual Reality (VR) und Augmented Reality (AR) in Sachen Verbreitung oder Marktgröße noch nicht mit klassischen Videospiele mithalten können, sind eine Vielzahl einflussreicher Firmen in diesem Bereich aktiv und arbeiten kontinuierlich an neuen Inhalten oder verbesserter Technologie. Obwohl beide Medien etwas mit Interaktion zu tun haben und der Inhalt immer zur Laufzeit von Prozessoren berechnet wird, unterscheiden sich VR und AR sowohl stark voneinander als auch von klassischen Videospiele. Allerdings sind die notwendigen Technologien für Videospiele wie Engines, APIs oder Chips schon recht nah an dem dran gewesen, was man für VR oder AR benötigt. Daher haben diese Medien die bestehenden Entwicklungsprozesse und Tools nicht grundlegend umgewandelt, sondern größtenteils nur erweitert.

Für das vollständige Eintauchen in eine andere Welt eignet sich VR am besten. Die Headsets werden um den Kopf geschnallt und umschließen die Augen vollständig – sieht man von Implementierungsdetails einmal ab, ist ein VR-Headset einfach

nur ein geschlossener, ohrumschließender Kopfhörer für die Augen. Man sieht nichts mehr von der realen Welt und sieht zu einhundert Prozent den virtuellen Inhalt. Das Gamepad, das an Videospielekonsolen zur Eingabe von Steuerbefehlen genutzt wird, wird hier meistens in zwei Teile geteilt und an die Hände geschnallt. Zusätzliche Sensoren wie Accelerometer oder Lage-sensor erfassen Bewegungen von Kopf über die Hände bis zu den Fingern. Je mehr dieser Bewegungen erfasst werden, desto höher bezeichnet man den Freiheitsgrad des Headsets. Ein Headset, das den Kopf mit drei Freiheitsgraden erfasst, erlaubt es Spielern, sich innerhalb der virtuellen Realität nur auf einer fixen Position umzudrehen. Headsets mit sechs Freiheitsgraden erfassen zusätzlich die translatorischen Bewegungen und Spieler können zusätzlich nach vorne oder zur Seite gehen.

AR ist wie die Antithese zu VR. Man kann es sich so vorstellen, als würde man mit einem offenen Kopfhörer in der Straßebahn Musik hören – die Klangkulisse der Straßebahn überlagert sich mit der Musik. Das passiert bei AR beim Betrachten der realen Welt mit einem zusätzlichen virtuellen Overlay. Die Herausforderung liegt hier darin, die beiden Welten sinnvoll miteinander zu vereinen.

Das wohl bekannteste AR-Videospiel ist *Pokémon Go*. Aber AR ist viel mehr als ein paar drüber gemalte Bilder auf einer Smartphone-Kamera, denn auch für AR gibt es seit ein paar Jahren sehr interessante Headsets beispielsweise in der Form der



Abb. 5: Screenshot aus *Super Mario Bros.*

HoloLens 2 von Microsoft oder die Light in Brillenform von Nreal. Vereinfacht gesagt, hat man zwei Brillengläser vor den Augen, kann also problemlos die gesamte Umwelt sehen und sieht aber auch auf den Brillengläsern ein virtuelles Bild. Optische Sensoren auf den Headsets erfassen die eigene Position in der Umgebung sowie die der Hände und Finger. Das ermöglicht nicht nur Hologramme wie jenes von Prinzessin Leila aus *Star Wars*, es erlaubt sogar mit Hologrammen zu interagieren. Aber all diese Worte nützen nur bedingt, denn AR ist wie die Matrix: Man muss es mit seinen eigenen Augen gesehen haben.

Bedeutung von Sound in AR und VR

So unterschiedlich AR und VR in der Nutzung auch sind, am Ende setzt man den Nutzern eine virtuelle Kamera auf den Kopf und gibt ihnen die Möglichkeit, sich frei zu bewegen. Während es in klassischen Videospiele noch möglich war, hin und wieder die Spieler auf etwas bestimmtes zu fokussieren, beispielsweise durch filmische Zwischensequenzen oder temporäres Erzwingen von Kamerabewegungen oder durch auffällig animierte Grafiken im User Interface, sind solche Methoden in AR und VR ein Garant für sofortige und heftige Übelkeits-, Schwindel- und Desorientierungsgefühle.

Eine plötzlich einsetzende Kamerafahrt kann in VR dazu führen, dass die Spieler umfallen oder sich erbrechen. Eine erzwungene Kameradrehung fühlt sich an, als würde jemand einem den Kopf herumreißen. Klassische User Interfaces, wie zum Beispiel die Gesundheitsanzeige der Spielfigur, müssen in die virtuelle Welt als Objekte mit physischer Wirkung integriert und können nicht zweidimensional „an den

Bildschirmrand“ platziert werden. Visuell muss man sich hier also zurückhalten. Doch da die Spieler in diesen Medien so frei agieren können, konzentrieren sie sich gerne auf alle möglichen Dinge, aber nicht auf das, was sie sollen. Diese Rahmenbedingungen haben dem Thema Sound in VR und AR eine große Bedeutung gegeben, denn egal wo man gerade steht und hinguckt, lässt sich ein plausibel wirkender Sound, der die Aufmerksamkeit der Spieler zurückgewinnt, verhältnismäßig einfach umsetzen.

Ortbarkeit in VR und AR

Eine präzise Ortbarkeit von Sounds hilft nicht nur bei der Fokussierung der Spieler, sondern trägt auch zur generellen Glaubhaftigkeit der virtuellen Welt bei. Vor der Existenz von VR und AR kamen in Videospiele aber zum allergrößten Teil nur klassische Lautstärke- und Panorama-basierte Ortungsverfahren für die Richtung des Sounds vor. Damit ein Geräusch hinter den Spielern auch von hinten wahrgenommen wurde, musste man also auch Lautsprecher hinter den Spielern aufstellen oder richtungsabhängige Filter verwenden, deren Nutzen aber relativ begrenzt war. Für Videospiele auf einem Bildschirm war das dennoch ausreichend.

Erst mit der Entwicklung von VR- und AR-Headsets wurde es notwendig, die Ortbarkeit generell zu verbessern, denn das Aufstellen vieler Lautsprecher um den Spielbereich ist aufwendig und teuer, von akustischen Optimierungen für die geeignete Wiedergabe der virtuellen Klangwelt ganz zu schweigen. Die für die Endnutzer kostengünstigere Lösung ist daher der Einsatz von Kopfhörern, mit der sich die

Video 3: *Super Mario Bros. Jump*
Variations: Subtile Variation von Klängen beugt bei Videospiele der Ermüdung des Gehörs vor. Hier passt sich der Sound in Klangcharakter, Länge und Tonhöhe an die Größe von Mario sowie der Höhe seines Sprungs an.



<https://vimeo.com/694444147/eab1b48c9b>

Ortung hauptsächlich mit Signalverarbeitung lösen lässt.

Ein wichtiger Grundbaustein der Ortung sind kopfbezogene Übertragungsfunktionen, sogenannte HRTFs (Head-Related Transfer Function). Diese bilden zum einen die Laufzeitdifferenzen zwischen den Ohren ab, zum anderen aber auch die frequenzabhängigen Amplitudenmuster, welche durch Beugungseffekte an Kopf und Schulter und richtungsabhängigen Reflexionsmustern der Ohrmuschel entstehen. Diese Eigenschaften können mit Filtern ausreichend beschrieben werden und bilden sowohl 360 Grad in der horizontalen Ebene (Azimuth) als auch 180 Grad in der vertikalen Ebene (Altitude) ab. In der Praxis werden Messungen von unterschiedlichen Richtungen einer Schallquelle als Impulsantworten gespeichert und zur Laufzeit anhand der Lage von Audio Source und Audio Listener die zur Richtung passenden Impulsantworten ausgewählt, verarbeitet und auf die Audio Source angewandt. Fertig ist der virtuelle Kunstkopfhörer!

Nicht unerwähnt bleiben sollten die Probleme, die bereits vom Kunstkopfhörer bekannt sind. Die Wiedergabe macht im Grunde nur auf Kopfhörern Sinn, derartige Formate sind also nicht besonders flexibel und klingen eher schrecklich bei Wiedergabe über Lautsprecher. Da man sich bei AR und VR allerdings bereits einen Bildschirm um den Kopf schnallt, kann man auch gleich noch einen Kopfhörer dazu packen. Die geringe Flexibilität wird hier also eher kein Problem.

HRTFs funktionieren am besten, wenn die Köpfe, Schultern und Ohren der Spieler nicht zu stark von den Personen bei der Messung abweichen. Das ist eher schwierig

umzusetzen, solange man nicht selber seine eigenen HRTFs erstellen lässt. Momentan bieten vereinzelt Spiele unterschiedliche HRTF-Profilen an, aus denen man durch Ausprobieren das für einen selbst passendste Profil auswählt. Solange es also keine Technologie gibt, eigene HRTFs auf einfache Art und Weise zu erstellen beispielsweise über einen 3D-Scan des Kopfes, wird die richtungsbezogene Lokalisation für manche Spieler passabel funktionieren und für andere nicht.

Spatialization in VR und AR

Neben der Richtung spielen aber auch räumliche Effekte eine große Rolle bei der Lokalisation von Schallquellen. Das ganzheitliche Einbeziehen der Umgebungseffekte wie Reflexionen, Verdeckung oder Abstrahlcharakteristik einer Schallquelle bezeichnet man als Spatial Audio und die Tools innerhalb von VR und AR als Spatializer. Zwar ist das prinzipiell nichts, was es nicht bereits in klassischen Videospiele oder Filmen gibt. Allerdings verkommt beispielsweise ohne die passenden umgebungsabhängigen Erstreflexionen die HRTF-basierte Richtungslokalisierung schnell zu einer In-Kopf-Lokalisation oder zumindest zu einer Zu-Nah-Am-Kopf-Lokalisation. Daher ist es nicht weiter verwunderlich, dass alle bekannten Spatializer wie Steam Audio, Oculus Audio SDK oder Google Resonance auch Erstreflexionen und Nachhall gleich mitliefern. Vor allem Steam Audio geht in diesem Bereich sehr weit und bietet einen vollkommen physikbasierten Ansatz, der sowohl raytracing-basierte Erstreflexionen und Verdeckung beherrscht als auch Luftschalldämpfung und dipol-basierte Abstrahlcharakteristiken der Schallquelle. Aber nicht vergessen, physikalisch korrekt sollte nicht mit bester Lösung gleichgesetzt werden.

Wie in Filmen und Videospiele ist auch in VR und AR das primäre Ziel, Intention und Inhalt zu unterstützen und nicht nur die Realität umzusetzen. Nur weil es jetzt viele neue Möglichkeiten gibt, muss man nicht alle für jeden Sound nutzen. Die HRTF-basierten Spatializer haben auch Schwächen und können bei bestimmten Geräuschen Artefakte oder starkes Phasing erzeugen. Am Ende zählt, was gut klingt und das bedeutet auch, dass man nicht jeden Sound durch einen Spatializer schicken muss.

Ortbarkeit und Spatialization im Forschungsprojekt Virtual History

Von den vorangegangenen Problemen mit der HRTF-basierten Richtungslokalisierung habe ich beim Forschungsprojekt *Virtual History* der Filmuniversität Babelsberg, welches das Forum Romanum aus der cäsarischen Epoche Roms nachstellt, ein 24-kanaliges Lautsprechersystem um den VR-Spielbereich aufgebaut und ein separates Rendering-System entwickelt.

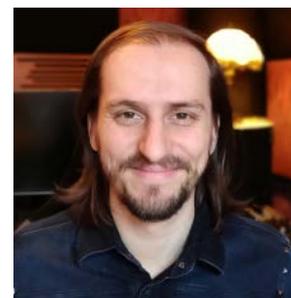
Middlewares und Engines haben üblicherweise eine Begrenzung auf vorgeschriebene Lautsprecher-Layouts und unterstützen in der Regel nicht mehr als 7.1 oder 7.1.4 bei Dolby Atmos. Daher wurde das Audio-Rendering um ein Plugin erweitert, welches über die Open-Source-Library JACK bis zu 56 Audioobjekte und ein 7.1-Bed an einen separaten Renderer weiterleitet, wo die Objekte mit VBAP (Vector Base Amplitude Panning) auf ein Lautsprechersystem mit acht Höhen- und acht Bodenlautsprechern gemischt werden. Einige besonders auffällige Geräuschquellen wurden um die raytracing-basierten Erstreflexionen von Steam Audio erweitert, was über verschiedene Orte auf dem Forum Romanum mit diversen Gebäudeformen zu markanten Echomustern führt.

Die Wiedergabe über ein um das Spielfeld gebautes Lautsprechersystem sorgte im direkten Vergleich zur umschaltbaren Kopfhörermischung für eine konsistentere Ortbarkeit von Geräuschquellen, steht aber unter kommerziellen Gesichtspunkten in einem ungünstigen Verhältnis von Aufwand, Nutzen und Kosten. Eine interessante Erkenntnis war, dass es bei Wiedergabe über Lautsprecher in VR notwendig ist, den Audio Listener für die Richtung auf die Mitte des VR-Spielfelds zu platzieren, sonst würde die gesamte Klangkulisse beim Bewegen innerhalb des Spielfeldes am eigenen Kopf hängen bleiben, was man auch als headlocked bezeichnet. Hingegen musste der Audio Listener für die raytracing-basierten Erstreflexionen an der VR-Kamera befestigt werden, damit sich die Reflexionszeiten an die Position des VR-Headsets anpassen. Auf der Sound-Design-Ebene gab es ein interessantes Problem, bei dem ein Sound durch den HRTF-basierten Weg oder den objektbasierten Weg zwar gut zu orten war, aber dadurch leider auch viel zu sehr an Breite eingebüßt hatte. Ein wesentliches Merkmal von HRTFs und dem Panning von Mono-Objektquellen ist, dass sie idealerweise gesehen zu einem ortbaren Punkt werden. Abhilfe hatte hier geschaffen, den Sound

in zwei Layer aufzuteilen und zwar in ein diffuses, breites Mehrkanal-Layer und ein klares Mono-Layer. Nur das Mono-Layer wurde dann spatialized, sodass der Sound vom Mehrkanal-Layer angedickt werden und sich sowohl auf Kopfhörer als auch auf Lautsprechern gut übertragen konnte. Für statische Geräuschquellen auf dem Forum Romanum wie raschelnde Bäume oder Büsche genügte es hingegen, mehrere Objektquellen nebeneinander zu platzieren. So war jede einzelne Quelle für sich genommen auch aus nächster Nähe eine dünne Punktquelle, die benachbarten Quellen fügten allerdings ausreichend Breite in den Gesamtklang hinzu. Übrigens war hier ein netter Trick, zwar bei allen Quellen das gleiche Wave-Asset zu verwenden, aber die Startposition pro Quelle zufällig über die Clip-Länge zu verteilen und die Abspielgeschwindigkeit zu variieren. Dadurch rauscht es von jeder Quelle anders, obwohl man nur ein einziges Asset verwendet.

Fazit

Die Sound-Design-Arbeit für interaktive Medien ist ein spannendes Feld für Tonschaffende mit vielfältigen Aufgabenbereichen, in denen sich die Grenzen des Möglichen kontinuierlich weiterentwickeln. Wem dieser Artikel Lust auf mehr bereitet hat, dem kann ich nur empfehlen, sich die genannten Tools einmal genauer anzusehen. Persönlich sehe ich viel Potenzial in den Sound- und Musik-Gestaltungsmöglichkeiten bei Virtual und Augmented Reality, aber auch bei prozeduralem Audio.



David Schornsheim ist freischaffender Komponist, Sound Designer und Entwickler für Games, Serious Games, VR und AR. Er arbeitet außerdem als Softwareentwickler an den Synthesizern und Effekten von u-he.