

PAQARITAMPU VR

Projektdokumentation

Team:

Christoph Painsipp: Projektleiter, Game design, Level design, Sound design, Audio implementation

Robin Unger: Programming

Anne Wahlmüller: Concept art, 2D art, Modeling, Texturing

Fabian Filipp: Sculpting, Texturing

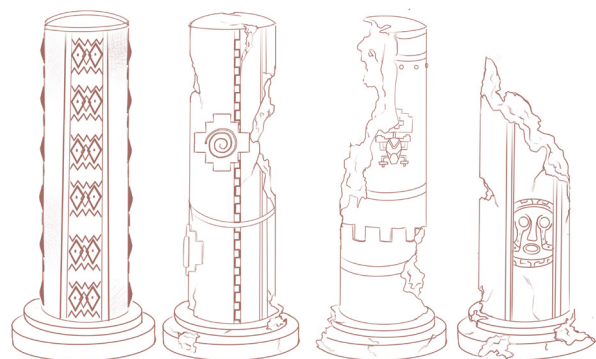
Konzept:

Das Konzept war sehr schnell gestaltet, da sowohl Christoph als auch Anne sehr große Horrorfans und Escape/Puzzle Gamer sind. Christoph wollte unbedingt ein VR-Spiel daraus machen, da vor allem Horrorspiele in VR einen großen Einfluss haben. Die Story dazu war schnell festgelegt. Durch die jahrelange Spielerfahrung stand auch sofort fest, dass ein Horrorgame mindestens im semirealistischen Bereich angesiedelt sein muss. Für das Gamedesign holte sich die Gruppe Ideen von erfolgreichen Spielen. Da die Geschichte auf Legenden der Inka basiert, wurde zuerst viel recherchiert, um nicht nur geschichtliche Informationen zu sammeln, sondern vor allem Ideen für den Stil bestimmter Objekte zu erhalten

Inhalt:

„Paqaritampu VR“ ist ein Horror - Puzzle Game. Die unwissentlich auserwählte Nachkommin des Sonnengottes Inti sucht mit ihren Freunden in einem Dschungel auf Hinweise ihrer Geschichte. Wegen eines heftigen Sturms geraten sie in eine Höhle. Die Freunde beschließen weiter hinein zu gehen, da sie sonst nichts tun können während der Sturm wütet. In der Höhle passieren die Dinge Schlag auf Schlag und bald ist die Protagonistin (Koya Evans) alleine und sucht ihre Freunde und einen Weg ins Freie.

Das Spiel ist für die Oculus Quest designt. Dadurch hat man die Möglichkeit, sich selbst zu bewegen und zu gehen, um somit ein immersives Gefühl zu vermitteln. Man kann das Spiel aber natürlich auch mit einem Teleport spielen, da uns bewusst ist, dass nicht jeder 7x7 Meter Platz zur Verfügung hat.



Bei den Konzepten wurde darauf geachtet, reale Symbole von den Inkas zu verwenden.

Herangehensweise:

Da das ganze Team noch nie ein VR Spiel kreiert hatte, mussten sich alle vorerst generell mit den „Aufgaben“ von VR beschäftigen. Wir wussten, dass die Oculus Quest einen Android Prozessor hat, weswegen uns bewusst war, dass wir bestimmte Dinge wie Übermengen an Particles oder zu hochaufgelöste Meshes nicht verwenden konnten. Christoph und Anne machten am Anfang eine Einteilung für alle Mitglieder, damit jeder Bescheid wusste, welche Aufgaben er/sie hatte. Des Weiteren wurde für Anne und Fabian am Anfang festgelegt, wer welche Assets übernimmt; beziehungsweise, welche wichtigen Assets zu welchem Zeitpunkt fertig sein müssen.

Nachdem die wichtigsten Konzepte fertig waren, begannen wir mit den ersten Programmierversuchen mit den Toolkits von der Oculus Quest. Danach fingen wir mit dem Sculpten und Modeln der wichtigsten Assets an. Gleichzeitig begannen wir dann auch mit dem Leveldesign. Hier standen wir schon vor den ersten Hürden: Das Leveldesign auf nur 7x7 Meter war sehr schwierig zu gestalten. Des Weiteren kämpften wir mit dem „Ab- und Anschalten“ der Räume, um den SpielerInnen das Gefühl zu vermitteln, dass diese „immer weiter“ gehen, obwohl sie lediglich auf den 7x7 Metern unterwegs sind. Dies brachte einige Schwierigkeiten mit sich, da es fast unmöglich war, ein Level für nur 7x7 Meter spielbar zu machen und es sich gleichzeitig für die SpielerInnen gut anfühlt. Es wurden sehr viele Arbeitsstunden darin investiert, das Game so zu designen, dass es weiterlaufend war. Die Übergänge zwischen den einzelnen Abschnitten mussten halbwegs flüssig und kaum bemerkbar gestaltet werden und dennoch genug Platz für die SpielerInnen lassen. Gleichzeitig überlegten wir, welche Rätsel und Hinweise, die zur Lösung wichtig sind, eingebaut werden könnten. .

Die ersten vier Levelabschnitte wurden vorskizziert, damit wir beim Bauen des Levels in 3D Anhaltspunkte hatten. Da uns aber dann schnell bewusst wurde, dass diese Art von Leveldesign sehr schwer umsetzbar ist, versuchten wir nun, die Räume offener und zugänglicher zu gestalten. Wir hatten zum Glück keine Probleme mit dem Einfügen von Assets und konnten somit die Höhle sehr schnell und effektiv mit ihnen ausschmücken, um so eine bestimmte Atmosphäre zu gestalten. Das Designen eines Levels war aber sehr schwierig und dauerte meist sehr lang. Besonders anfangs war das Leveldesign sehr fordernd, da wir aufgrund der Platzbeschränkung auf Wände, die nur aus Planes designed wurden, zurückgegriffen hatten.

Leider war eines der größten Probleme, dass es zur Oculus Quest und das zum Designen eines Games für die Quest sehr wenige Informationen gab, vor allem für den Audiodbereich. Somit hatten wir des Öfteren Probleme, bei denen es keinerlei Unterstützung gab und wir letztendlich meist anhand von „trial and error“ arbeiten mussten. Das Audiosystem musste ein paar Mal geändert werden, da es immer wieder zu Komplikationen kam. Wie bereits erwähnt, gibt es wenig Dokumentation zu der Oculus Quest, vor allem für Audio. Daher wurde das Audiosystem zuerst von FMOD mit Oculus auf FMOD mit Steam Audio und dann auf FMOD mit Ressonance Audio gewechselt. Letztendlich bewegten wir uns ganz von FMOD weg und haben das Audio auf Oculus ONSP SDK in Unity Native gewechselt. Aufgrund der Art, wie das Game gebaut ist, und die Abschnitte, welche sich an- und abschalten, programmiert sind, wurde bei Steam Audio ständig die baking-Cache für den Pre-Computed Reverb gelöscht. Bei Ressonance Audio war es lediglich möglich, über ein Shoebox Modell zu arbeiten, welches in unserem Kontext genauso wenig funktionierte. Dies war hauptsächlich darauf zurückzuführen, dass der Reverb in einer Shoebox lediglich für die Sound Sources innerhalb dieser generiert wurde und vom Listener auch nur dann wahrnehmbar war, sobald dieser die Shoebox betreten hatte. Stellte man also zwei Shoeboxes nebeneinander, so hörte man den Reverb erst gar nicht und dann plötzlich mit drastischem Cut beim Betreten der anderen Shoebox. Auch das Baken vom Reverb mit Ressonance Audio verursachte dieselben Probleme wie Steam Audio zuvor.

Da in Präsentationsvideos von Oculus Connect die Real-Time Audio Propagation der Oculus SDK auch als funktionell für Android Anwendungen angepriesen wurde, fiel der Entschluss, wieder auf Oculus umzusteigen. Erst durch gegenlesen der verschiedenen Dokumentationen der Oculus SDK für die verschiedenen Plattformen wurde festgestellt, dass die Audio Propagation bei Verwendung von Fmod mit Unity nicht unterstützt wird. Es wurde in den Dokumentationen zwar nicht explizit aufgewiesen, dass dies der Fall war, es wurde die Audio Propagation Dokumentation jedoch bei Fmod-Unity, im Gegensatz zu allen anderen Dokumentationen, einfach weggelassen. Aufgrund dessen entschlossen wir uns dazu, von Fmod wegzugehen und das Audio, Unity intern, mit der nativen Oculus Spatializer SDK zu verbauen. Ein Umbau auf Wwise kam auch nicht in Frage, da dafür schlichtweg die Zeit viel zu knapp wurde. Daher mussten auch alle bereits fertigen Scripten für das Triggern und Implementieren von Sound zu bestimmten Events und Stellen im Game umgeschrieben werden, was bereits einiges an Zeit beanspruchte. Auch alle Sound Sources und Ambiences sowie Ambience Switches mussten neu implementiert, eingestellt und gemischt werden. Zu diesem Zeitpunkt hatten wir es endlich geschafft, einen funktionierenden Reverb im Game zu etablieren, auch Occlusion funktionierte, wenn auch nur anhand eines einfachen Line-of-Sight Prinzipes, wodurch der Schnitt zwischen hörbaren und nicht hörbaren Audio Sources sehr abrupt klang. Doch plötzlich traten andere Schwierigkeiten auf. Audio Sources schienen occludiert zu sein, obwohl zwischen Listener und Klangquelle keinerlei Hindernis lag. Doch machte man einen Schritt nach links oder bewegte man den Listener zwei Meter nach unten, hörte man die Sound Source plötzlich von 0 auf 100. Selbst Einstellungen an den Attenuations änderten hier nichts. Es schien einfach so, als wäre eine unsichtbare Wand zwischen dem Listener und der Klangquelle, auch wenn dies nicht der Fall war. Ebenfalls änderten die bereits richtig platzierten Sources beim Starten des Play-Modus plötzlich ihre Position aus unersichtlichen Gründen. Zu guter Letzt wollten wir unseren ersten Build mit dem neuen Audiosystem an Tester übergeben, doch plötzlich hatte der Build keinerlei Audio mehr, obwohl es im Play-Mode funktioniert. Hierbei sind wir noch am De-Buggen und Robin versucht herauszufinden, was genau dieses Problem verursachen könnte.

Auch programmiertechnisch stießen wir immer wieder auf Komplikationen, die wir zu lösen hatten, ohne einen Anhaltspunkt zu haben, da auch hier die Dokumentaion zur Quest sehr schwach war. Bis zum Schluss traten immer wieder neue Probleme auf, die nur umständlich lösbar waren. Angefangen vom Roomscale und Real-Time-Walking, wozu es keine Referenzen gab, bis hin zum Programmieren der Levels.