Bild: CC-BY Jonathan Williamson

Ozean-Simulation in Blender 2.6x

Blender hat seit Version 2.61 eine Ozean-Simulation als Modifier, die auf dem Houdini Ocean Toolkit basiert. Die Finanzierung der Integration gelang wie so vieles in letzter Zeit über eine Crowdfunding-Kampagne. Digital Production stellt die wichtigsten Funktionen der Simulation vor, zeigt einige Anwendungsbeispiele abseits der Ozeane und sprach mit dem Initiator des Fundraisers.

von Gottfried Hofmann

Das Houdini Ocean Toolkit (HOT) ist ein Open Sourcce-Projekt, das inzwischen auch für 3D Studio Max sowie Blender portiert wurde. Das HOT erlaubt die realistische Simulation der Wasseroberfläche eines tiefen Ozeans. Sprich Wellenbrecher und andere Phänomene, die am abflachenden Wasser an der Küste auftreten, können damit nicht simuliert werden, wohl aber die Wellenbewegungen auf dem offenen Meer.

Die Portierung des HOT zu Blender ist ein Paradebeispiel, wie Studios ihre indivduellen

Displace – Ein stark unterteilter Text von der Ozean-Simulation verzerrt...

Bedürfnisse durch Beteiligung an der Entwicklung von Blender befriedigen können. Ursprünglich wurde der Code des HOT von Promotion Studio für den Kurzfilm "Lighthouse" nach Blender 2.49 portiert. Dies geschah über einen freiberuflichen Entwickler der durch einen festangestellten unterstützt wurde.

Nach Fertigstellung des Kurzfilms wurde der Code für die Portierung veröffentlich und einige Nutzer kompilierten spezielle Blender-Versionen, die die damalige Ocean-Simulation enthielten. Doch in den Hauptentwicklungszweig konnte sie aus verschiedenen Gründen nicht Einzug halten.

Als die Version 2.5x von Blender veröffentlicht wurde, entschloss sich ein Blender-Nutzer, der nicht mit dem Studio in Verbindung steht, eine Portierung auf die neue Version per Crowdfunding zu finanzieren. Dabei sollte auch das Design der Integration so abgeändert werden, dass die Simulation auch im Hauptentwicklungszweig landen könnte. Die Kampagne war so erfolgreich, dass auch noch Geld für die Erweiterung der Funktionalität übrig war. Digital Production sprach mit dem Initiator der Kampagne, Todd McIntosh.

Die Früchte der Arbeit

Seit Version 2.61 ist die Ocean-Simulation nun in Blender als Modifier integriert. Das bedeutet, sie kann zum Verformen eines bestehenden Meshes eingesetzt werden. Als zusätzliche Option wird angeboten, ein vorhandenes Mesh komplett durch einen simulierten Ozean zu ersetzen. Die Änderungen sind dabei nicht-destruktiv bzw. reversibel und können mit anderen Modifiern bzw. Simulationen wie Dynamic Paint kombiniert werden. Einen einfachen Ozean können Sie erstellen, indem sie einem beliebigen Mesh einen ‚Ocean'-Modifier hinzufügen. Dieser ist von vorner herein auf ‚Generate' eingestellt. Das bedeutet, dass das Mesh komplett durch den Ozean ersetzt wird. Eine Änderung auf ‚Displace' lässt die Simulation auf die bestehenden Vertices des Meshes einwirken. Diese Option ist dann praktisch, wenn der Ozean nicht auf eine flache Fläche, sondern z.B. einen Text angewandt werden soll oder wenn man die Anzahl der Faces aus Performance-Gründen genau kontrollieren will.

Eine weitere wichtige Option ist "Time". Diese gibt an, an welchem Zeitpunkt (in Sekunden) man sich in der Simulation gerade befindet. Wenn man diesen Parameter animiert, animiert man also auch die Simulation und erhält einen wogenden Ozean.

Die Wellen der Simulation schließen in X- und Y-Richtung an, sind also ‚seamless'. Dies lässt sich entweder ausnutzen wenn man aus der Simulation eine Displacement-Map bäckt (dazu später mehr) oder indem man die Optionen "Repeat X/Y" nutzt. Dabei werden automatisch Kopien erzeugt, die direkt an das Mesh anschließen und damit für zusätzliche Ozean-Oberfläche ohne Extra-Berechnungen sorgen.

Weiterhin kann Schaum berechnet werden, wie er an den Spitzen von Wellen entsteht. Dieser wird in den Vertex-Farben des Meshes gespeichert. Hierbei muss beachtet werden, dass ein Wert von 0 für die normale Menge an Schaum steht und negative Werte für weniger, positive Werte für mehr Schaum sorgen.

Auch in anderen Programmen einsetzbar

Die Displacement-Map, die Schaum-Textur und auf Wunsch auch die Normalen können "gebacken" werden. Das bedeutet, dass sie als Bildsequenz im OpenEXR-Format auf die Festplatte ausgegeben werden. Dadurch können die Ergebnisse der Ozean-Simulation auch in anderen Programmen eingelesen werden.

Für den Blender-internen Render wird zudem automatisch eine Ocean-Textur erstellt, die für das Shading genutzt werden kann. Der Cycles-Render kann auf diese Texturen noch nicht zugreifen, für den normalen Anwendungsfall dürfte aber nur die Schaum-Textur notwendig sein, und diese ist wie schon erwähnt in den Vertex-Farben gespeichert, die über die "Attribute"-Node auch von Cycles eingelesen werden können.

Werden die Wogen geglättet?

Die restlichen Einstellungsmöglichkeiten der Ozean-Simulation beziehen sich auf die zugrundeliegende Physik bzw. das Verhalten der Wellen und sollen im Folgenden vorgestellt werden.

Mit der Option „Resolution" lässt sich die Auflösung bzw. Genauigkeit der Simulation beeinflussen. Wenn die Simulation unter „Generate" betrieben wird, wird dabei auch automatisch die Anzahl der Vertices angepasst, sprich die Auflösung der Wellen wird mit höheren Werten immer feiner. Höhere Werte bedeuten aber auch eine längere Simulationszeit. Man beachte, dass dabei die

Generate – Wählt man stattdessen „Generate" (was die Voreinstellung ist), so wird das gesamte Mesh durch eine Plane ersetzt, die bereits die passende Unterteilung für den Ozean aufweist und schon entsprechend deformiert ist.

Projektplan per Fundraiser

Todd McIntosh hat die Crowdfunding-Kampagne für die Ozean-Simulation in Blender 2.5 organisiert. Ursprünglich war diese ein Port des Houdini Ocean Toolkit für Blender 2.49, ziel der Kampagne war eine Portierung auf Blender 2.5 und aufgrund des Erfolgs auch eine Erweiterung der Funktionalität. Todd leitet außerdem das Web-Entwicklungs und -Design-Studio „Resonance Media" aus Markham, Ontario in Kanada.

DP: Erzähl uns ein wenig über das „Save the Ocean-Sim"-Projekt. Worum ging es dabei, woher hattest du die Idee etc.?

Todd McIntosh: Der Grund für das Projekt war ein animierter Kurzfilm, den ich in meiner Freizeit erstelle und der „Reign of Fury" heißt. Der Film basiert lose auf einer Epoche zu Beginn des Zeitalters der Vikinger und beinhaltet natürlich Vikinger sowie Nordumbrische Mönche. Ich wollte den Film so realistisch halten wie es meine Fähigkeiten erlauben und dazu gehören einige glaubwürdig aussehende Szenen mit Schiffen auf dem Ozean. Ich suchte nach Methoden für das erstellen von realistischen Wellenbewegungen in Blender und fand nur wenige brauchbare. Dazu gehörte der Port des Houdini Ocean Toolkit für Blender 2.49. Der wurde von der Agentur „Red Cartel" (James Neale) aus Australien finanziert und kommerziell für den Kurzfilm „Lighthouse" in 2009 eingesetzt. Realistische Ozean-Wellen mit scharfen Spitzen sind nicht wirklich möglich ohne die spezifischen Muster, die der Algorithmus des HOT generiert. Davor konnte man in Blender nur die prozeduralen Standard-Texturen von Blender nutzem um das Displacement für die Wellen zu erstellen. Die verschiedenen Methoden produzierten nie die scharfen Wellenberge, die für eine stürmische See notwendig sind, nur relativ ruhige und glatte Oberflächen.

Kai Kostack hatte eine spezielle Version von Blender 2.49 mit integrierter Ozean-Simulation erstellt, aber ich wollte eine Version, die auf meinem Mac nativ lief. Und ich wollte außerdem, dass das Tool in Version 2.5 (die damals aktuelle Version von Blender) funktionieren würde. Zuerst habe ich den Entwickler Matt Ebb kontaktiert um einen Kostenvoranschlag zu erhalten. Ich wollte sehen welche Investition für einen Port auf Blender 2.5 notwendig sind und sie hielten sich in Grenzen. Obwohl ich ihn mit meinen eigenen Geldern nicht anheuern konnte dachte ich mir, es muss doch möglich sein ein paar andere Blender-Nutzer zu finden damit wir uns die Kosten teilen könnten. Also habe ich im größten Blender-Forum Blenderartists.org eine Abstimmung platziert. Nachdem sich herausstellte, dass mindestens 16 Leute das Vorhaben unterstützen würden erstellte ich die Webseite SaveTheOceanSim.com und veröffentlichte eine Ankündigung auf dem Community-News-Portal PlanetBlender.org.

DP: Welche Rolle spielte die Blender-Gemeinde für den Erfolg des Projekts?

Todd McIntosh: Ohne die Blender Community wäre kein einziger Dollar für das Projekt zusammengekommen. Es war das Feedback und Interesse der Blender-Gemeinschaft, das mich darauf brachte den Fundraiser ins Leben zu rufen. Die Reaktionen waren überwältigend. Die ursprünglich anvisierte Summe hatten wir binnen 24 Stunden zusammen. Also konnte ich Matt beauftragen, den alten Code der Ocean Sim auf 2.5 zu portieren. Was aber viel aufregender war - die Leute waren so begeistert von der Erwartung, in Blender bessere Ozean-Szenen erstellen zu können dass die Gelder weiter flossen. Die Kampagne lief für rund 2 Monate und am Ende hatten wir 2.400 US-Dollar zusammen, abzüglich rund 250 US-Dollar an Paypal-Gebühren. Das erlaubte uns das Tool zu verbessern und zusätzliche Features hinzuzufügen. Ein Beispiel ist die Schaum-Maske, die in Zukunft hoffentlich genutzt werden kann um dynamisch Partikel für Ozean-Spritzer zu erstellen.

DP: Gab es auch Kritik und wie seid ihr damit umgegangen?

Todd McIntosh:: Direkt habe ich keine Kritik erhalten aber es gab da ein Posting auf Blendernation das für einige Diskussion gesorgt ob ich den Fundraiser durchführen soll oder nicht. Das ich die Gelder direkt annahm und nicht z.B. eine registrierte gemeinnützige Organisation hat einigen Bauchschmerzen bereitet. Das sind natürlich legitime Kritikpunkte aber letztendlich ging es bei dem Projekt darum etwas zu schaffen was sonst keiner anfassen wollte. Ich konnte die Meisten dadurch beschwichtigen, dass alles so transparent wie nur möglich ablief. Ich erstellte regelmäßig Posts auf dem Blog und schickte an alle Spender E-Mails. Das Feedback der Spender war ausschließlich positig.

DP: Bist du mit den Resultaten zufrieden und hast du die Ozean-Simulation inzwischen selbst eingesetzt?

Todd McIntosh:: Mit dem Ergebnis bin ich sehr zufrieden, aber wir warten noch bis Partikel-Emissionen über Dynamic Paint fertiggestellt sind. Und wir hoffen dass es einen Weg geben wird, über die Schaum-Maske Partikel für Ozean-Sprizer generieren zu können. Damit und den Wellen für Dynamic Paint, an denen Miika Hämäläinen gearbeitet hat als Teil seines Dynamic Paint-Projektes denke ich, dass wir auf dem Weg zu einer anständigen, nicht auf Fluid-Simulationen basierten Ozean-Werkzeug in Blender sind.

DP: Solltest du wieder ein bestimmtes Feature in Blender benötigen, würdest du dann wieder einen Fundraiser durchführen?

Todd McIntosh: Ja würde ich, sofern es sich um ein Feature handelt das ich wirklich für mein Filmprojekt benötige. Obwohl ich nicht die Programmierarbeit bei dem Projekt durchgeführt habe hat es mir immer noch einiges an Zeit gekostet den Fundraiser aufzusetzen, zu bewerben und zu managen. Das großartige an Blender und Open Source im allgemeinen ist, dass jeder neue Tools für Blender schreiben kann. Oder man bezahlt einen Entwickler dafür. Natürlich haben Ton Roosendaal und Kern-Entwickler ein Mitspracherecht, ob Dein Tool in die Hauptversion von Blender gelangt. Aber selbst wenn nicht, die Möglichkeit, bei einer Produktion auftretende Probleme über eigene Entwicklungen zu lösen ist wirklich herausragend.

DP: Wie siehst du die Zukunft von Crowdfunding für Open Source im Allgemeinen und für Blender im Speziellen?

Todd McIntosh: Nun, ich bin sehr froh über den „Blender Development Fund", den Ton vor einiger Zeit aufgesetzt hat. Ich denke er wird helfen, Blender einen beständige Unterstützung zu geben bei der Entwicklung von größeren Zielen wie der Render Engine Cycles oder zum Beispiel die Integration der Open Shading Language (OSL). Was mich außerdem begeistert ist jene Freiheit die es jedem erlaubt, entweder selbst neue Werkzeuge zu erstellen oder über eine Crowdunfing-Kampagne mit der Unterstützung von Anderen das Tool zu bekommen, das sie selbst begeistert. Die Zukunft von Blender sieht rosig aus!

Schaum-Maske – Die Informationen für die Schaummaske werden in den Vertex-Farben gespeichert. In Cycles kann man darauf über die Attribute-Node zugreifen.

Auflösung – Über den Wert „Resolution" wird die Auflösung der Simulation eingestellt. Von Links nach Rechts bei ansonsten identischen Einstellungen: 7, 14, 21.

Räumliche Größe – Über den Wert „Spatial Size" wird die Größe der Fläche, die von der Simulation eingenommen wird, bestimmt. Von Links nach Rechts bei ansonsten identischen Einstellungen: 25, 15, 5.

Form und Größe der Wellen verändert wird. Es ist also nicht möglich, eine Simulation zuerst in niedriger Auflösung zu erstellen und wenn das Aussehen passt über eine Erhöhung der Auflösung lediglich Details hinzuzufügen. Die Einstellung „Spatial Size" hingegen ändert die Größe der Simulation. Im Falle von „Generate" wird dabei automatisch das Mesh angepasst.

Die Einheit sind hier Blender-Units. Dabei sei zu beachten, dass sich andere Einstellungen dann relativ zur Größe stärker auswirken. Für einen Swimmingpool benötigt man daher andere Einstellungen als für den Atlantischen Ozean.

Über „Choppiness" lässt sich die Schärfe der Wellenspitzen steuern. Intern ändert dieser Wert die Verzerrung in X- und Y-Richtung, was dazu führt, dass bei zu hohen Werten Artefakte von sich überlappenden Flächen auftreten können.

„Scale" hingegen ändert die Stärke in alle Richtungen, was dazu führt, dass die Wellen größer werden, aber wieder die Gefahr von Artefakten mitbringt.

„Smallest Wave" ist eine weitere Möglichkeit, die Feinheit der Simulation zu steuern. Mit diesem Wert wird eingestellt, wie klein die Wellen werden dürfen. Beim setzen von besonders großen Werten lässt sich die Simulation, die ja seamless ist, auch für wallende, abstrakte Hintergründe einsetzen.

„Wind Velocity" ist ein weiterer Wert, der mit Vorsicht zu genießen ist. Grundsätzlich wird über diesen die Geschwindigkeit des Windes reguliert, der über die Wasseroberfläche weht und dabei für Wellen-Turbulenzen sorgt. Dieser ändert dabei aber nicht die Größe der Wellen sondern nur ihre

Bild: CC-BY David Zerba

Ocean Simulation – Ein Beispiel für ein Setup der Ozean-Simulation.

Böigkeit – Über den Wert „Choppiness" lässt sich die Schärfe der Wellenspitzen ändern, allerdings zum Preis von Artefakten bei zu hohen Werten. Von Links nach Rechts bei ansonsten identischen Einstellungen: 1, 2, 3.

Wie groß darfs denn sein? Über den Wert „Smallest Wave" wird die minimale Größe der Wellen gesteuert. Von Links nach Rechts bei ansonsten identischen Einstellungen: 0, 0.5, 1.

Windgeschwindigkeit – Über den Wert „Wind Velocity" lässt sich das Verhalten der Wellen ändern, nicht jedoch deren Größe. Von Links nach Rechts bei ansonsten identischen Einstellungen: 3, 6, 18.

Ausrichtung – Über den Wert „Alignment" wird eingestellt, wie stark die Wellenbewegung der Windrichtung folgt, wobei in Zusatzoptionen die Windrichtung selbst und die Stärke der Dämpfung von Wellen aus anderen Richtungen eingestellt werden kann. Von Links nach Rechts bei ansonsten identischen Einstellungen: 0, 3, 10.

Bewegung. Sprich wer einen Ozean-Sturm simulieren will, sollte neben dem Wert für die Windgeschwindigkeit auch die Größe der Wellen erhöhen während für eine fast windstille Oberfläche der Wert entsprechend gesenkt werden sollte, sonst drohen Artefakte. Ein weiteres Beispiel, wie die einzelnen Einstellungen der Simulation untereinander interagieren.

„Alignment" regelt, wie stark die Wellen von der Richtung des Windes beeinflusst werden. Die Richtung selbst kann dabei über „Direction" gesteuert werden und der „Damping" regelt, wie stark Wellen abgedämpft werden, die sich entgegen der Windrichtung bewegen.

Fazit

Mit der Ocean-Simulation hat in Blender ein weiteres praktisches Werkzeug Einzug gehalten, mit den Ecken und Kanten die man auch von diversen anderen Simulationen her kennt. Dank der Möglichkeit, die Ergebnisse zu exportieren, kann es auch genutzt werden um einen Ozean für die Weiterbearbeitung in anderen Programmen zu erstellen, für die das HOT noch nicht portiert wurde. Wie z.B. Cinema4D. › ei

Gottfried Hofmann hat an der FAU Erlangen-Nürnberg Informatik studiert. Er arbeitet als Freelancer im VFX-Bereich sowie als Trainer für die freie 3D-Software Blender. Als freischaffender Autor schreibt er für Fach- und Computerzeitschriften. Er hat zahlreiche Blender-Tutorials verfasst, u.a. für CG Tuts+ und CG Cookie. Weiterhin betreibt er die Webseite www.BlenderDiplom.com, auf der Blender-Tutorials in deutscher und englischer Sprache zur Verfügung stehen.