



Einführung: Faszination 3D

Sicher habt ihr schon Anaglyphenbilder gesehen, vielleicht wart ihr auch schon mal in einem 3D-Kino. Jetzt habt ihr die Möglichkeit, eure eigenen Anaglyphenbilder direkt aus Blender zu erstellen, auch Filme sind kein großes Problem. Ihr braucht nur Blender, Kreativität und Geduld...

Leider muss für so ein Bild eure Szene zweimal gerendert werden, und die Verarbeitung der Bilder zum fertigen Anaglyphenbild dauert auch noch etwas. Wenn ihr also einen Film in 3D plant, solltet ihr einen starken Rechner und viel Zeit haben :)

Während ich dieses Tutorial schreibe ist die Version 1.3 aktuell.

So funktioniert's

Ihr erstellt eure Szene und startet das Script aus dem Scriptmenü (Gruppe Misc). Die Standardeinstellungen erzeugen ein Schwarzweißbild, das könnt ihr aber mit einem Klick auf RGB oder BW ändern. Außerdem müsst ihr wahrscheinlich den deltaX Wert anpassen. Falls ihr einen Film drehen wollt, stellt noch Start- und Endbild ein.

Zusätzlich könnt ihr noch wählen, ob ihr das rechte und linke Bild rendern wollt (Right und Left) und ob ihr die Bilder zusammensetzen wollt (Join). Diese Optionen zu verwenden kann sinnvoll sein wenn ihr euer Projekt in verschiedenen Etappen rendern oder die Bilder noch nachbearbeiten wollt. Falls ihr nachbearbeitet, solltet ihr die Dateinamen beibehalten, vor allem die Zusätze `_right_` und `_left_`, sonst findet das Script die Bilder nicht.

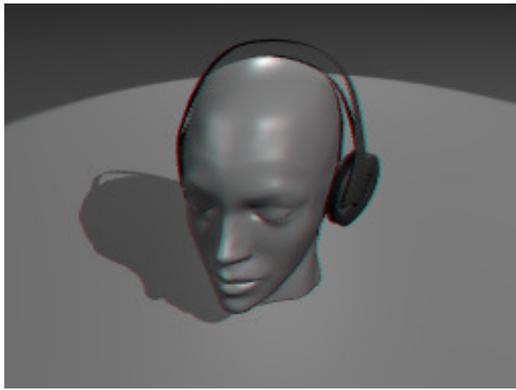
Jetzt müsst ihr nur noch rendern. Die Bilder werden automatisch am eingestellten Ort gespeichert, welcher das ist könnt ihr im Konsolenfenster nachschauen (Standard-Renderpfad, falls eingestellt). Dort seht ihr auch die Gesamtrenderzeit inklusive Verarbeitung. Wenn das Bild fertig ist, wird es außerdem angezeigt, ihr könntet es dann auch noch einmal speichern. Dann aber Achtung: das Script stellt den Dateityp auf Targa Raw um! (Das liegt daran, dass ich mit diesem Format am Besten umgehen kann.) Bei Animationen wird nicht jedes einzelne Bild angezeigt, sondern der fertige Film abgespielt.

Die Einstellungen des Scripts werden in der `.blend` gespeichert, bevor ihr rendert werden die aktuellen Einstellungen nochmal extra im Profil `** last render **` gespeichert. Beim Laden und Speichern müsst ihr nur aufpassen, dass ihr im Menü das richtige Profil ausgewählt habt. Um ein neues Profil anzulegen wählt einfach `** Add New **` im Menü, gebt im Textfeld daneben den Namen ein und speichert.

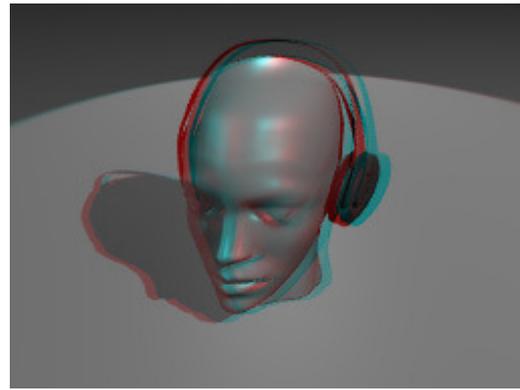
Der deltaX Wert

Den Abstand zwischen den Kamerapositionen kann man als Abstand zwischen den virtuellen Augen sehen. Ich hab mal irgendwo gelesen, dass dieser Abstand etwa 6 bis 8 cm betragen sollte, was dem normalen Abstand zwischen den menschlichen Augen entspricht. Das ergibt einen Wert von $\text{deltaX} = 0.06$ für 6 cm.

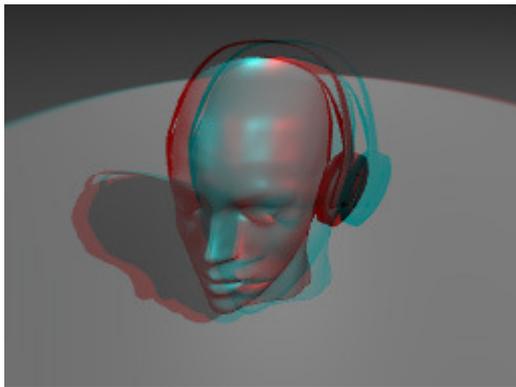
Eure Einstellungen hängen aber stark von den Größenverhältnissen eurer Szene ab, die Rendermethode hat ein deutliches Gewicht, und auch die Größe des gerenderten Bildes kann einen Einfluss auf den besten Wert haben. Bei kleinen Bildern kann der Wert etwas größer sein als beim endgültigen Ergebnis, da wegen der geringeren Auflösung des kleinen Bildes Details verloren gehen und beim großen Bild die blauen und roten Ränder vom Gehirn nicht mehr so gut weggerechnet werden können.



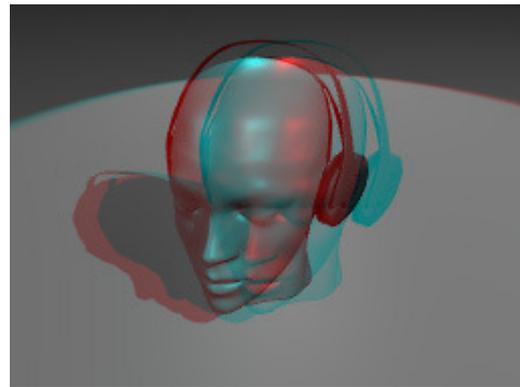
deltaX = 0.02



deltaX = 0.10



deltaX = 0.20



deltaX = 0.30

Zu den Beispielbildern:

Bei Bild 1 ist deltaX eindeutig zu klein, der 3D-Effekt kaum zu sehen.

Bild 2 kommt mir ideal vor, Bild 3 ist auch noch relativ gut.

Bei Bild 4 finde ich den Wert zu groß, der 3D-Effekt kommt zu stark raus. (Ich hab das mal größer gerendert: man kriegt fast einen Krampf in den Augen, wenn man das Bild anschaut)

Fazit: ausprobieren :)

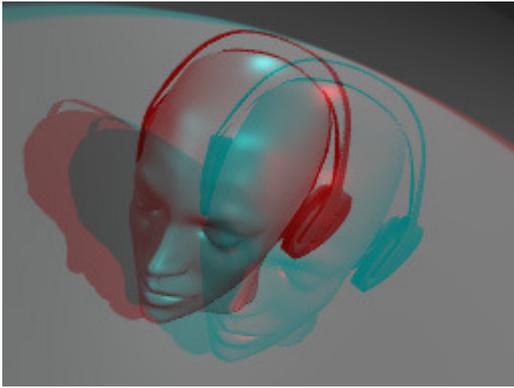
Correct Cam

Diese Funktion erzeugt ein Empty, das als Träger (wird als "parent" bezeichnet) für die Kamera dient. Das ist sinnvoll, wenn die Kamera für eine Animation bewegt wird oder gedreht ist (Beispielbilder).

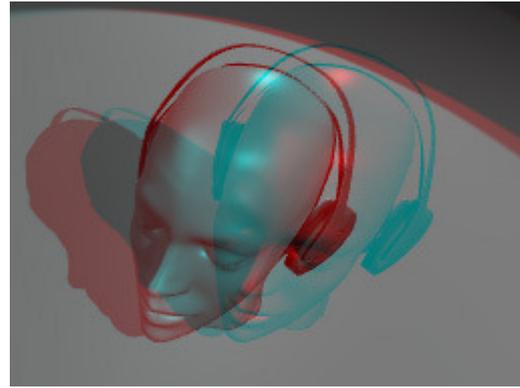
Je nach Drehung kann das Ergebnis falsch aussehen: das Script verschiebt die Kamera immer entlang der X-Achse. Wenn die Kamera um 90° zur Seite gedreht wäre, könnte man höchstens einen Größenunterschied erkennen, nicht aber eine seitliche Verschiebung.

Das Empty definiert das Koordinatensystem der Kamera neu, sodass jetzt wirklich eine seitliche Verschiebung stattfindet. Voraussetzung ist allerdings, dass dann nur noch das Empty bewegt oder gedreht wird, die Kamera macht automatisch alle Bewegungen mit.

Falls ihr das Empty löscht, wird die Kamera zurück an den Ursprung gesetzt, auch die Drehung geht verloren. Wenn ihr das vermeiden wollt, wählt die Kamera und drückt ALT P und dann 2 (Clear and Keep Transformation). Erst jetzt könnt ihr das Empty löschen, die Kamera ist nicht mehr daran gebunden.



ohne Korrektur



mit Korrekturempty

Animation

Auch für Animationen ist es sinnvoll, ein Trägerempty zu verwenden. Wenn ihr die Kamera bewegt, wird für jedes Bild die Position berechnet und das Script kann sie nicht ändern, deshalb wäre kein Effekt zu sehen. Wenn ihr die Kamera mit IPOs animiert habt, klickt auf Correct Cam, die IPOs werden dann für das Empty verwendet.

Im Moment noch nicht unterstützt:

Wenn die Kamera schon ein parent hat, funktioniert's nicht, am Besten vorher das Empty einsetzen und dann das andere Trägerobjekt mit dem Empty verbinden.

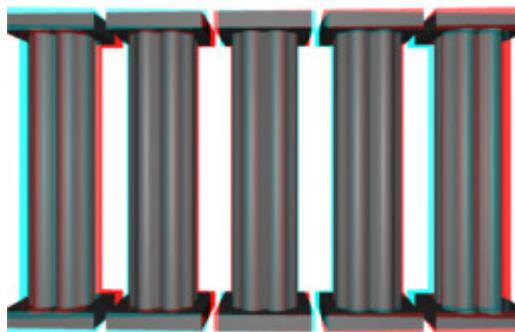
Zum Rendern: wenn ihr mal gestartet habt, könnt ihr das Script mit STRG C im Konsolenfenster abbrechen (funktioniert übrigens mit jedem Script), wenn ihr nur im Renderfenster ESC drückt, hält das Script nicht an. Dann solltet ihr eure Szene gleich laden, da das Script einige Einstellungen ändert. Deshalb: **bei Animationen immer vor dem Rendern speichern!**

Parallel oder nicht?

Es gibt eine endlose Diskussion, ob beim Rendern von Stereobildern die "Augen" parallel ausgerichtet oder auf einen Punkt fokussiert sein sollen.

Ich will mich nicht daran beteiligen, nur soviel: jede Methode hat ihre Vor- und Nachteile.

Im zweiten Fall sieht das Bild etwas anders aus, die rot/cyan Ränder hinter dem Fokuspunkt sind vertauscht. Der Effekt kommt vielleicht sogar besser raus, aber bei Objekten am Bildrand kann es Probleme geben, weshalb diese Methode umstritten ist. Wenn ihr aber nur in der Mitte rendert, werdet ihr vielleicht diese Methode bevorzugen, da deutlich weniger farbige Ränder entstehen und der Effekt gut rauskommt.

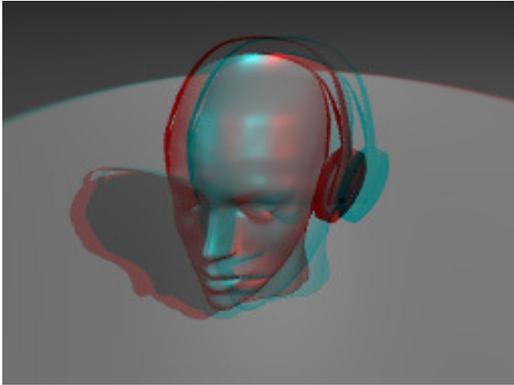


Problem der Fokusmethode

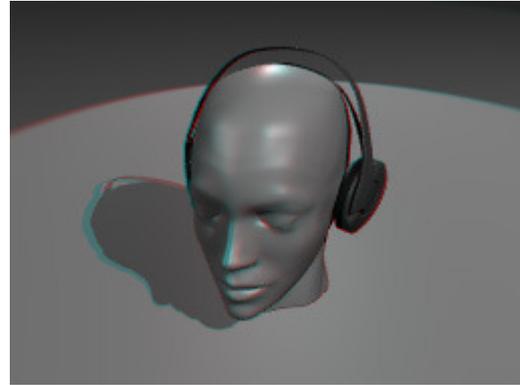
Oben solltet ihr deutlich sehen, was bei zu großem deltaX passieren kann: am linken Bildrand ist die Cyansäule eindeutig größer, am rechten die rote Säule. Das wurde irgendwo mal als Parallaxe bezeichnet, glaub ich. Dieses Problem besteht nur bei der Fokusmethode.

So kriegt ihr den Fokus hin:

Ich empfehle ein Empty als Fokusobjekt zu erzeugen, das könnt ihr frei bewegen und es stört das Bild nicht, ihr könnt aber auch jedes vorhandene Objekt verwenden. Also, wählt zuerst die Kamera und danach das Fokusobjekt dazu, dann drückt STRG T und 1 (TrackTo Constraint). Bei der Kamera erscheint dann in den Objekteigenschaften (F7) ein Constraint mit Namen Autotrack. Um den Fokus zu entfernen, löscht ihr entweder dieses Constraint oder das Fokusobjekt oder stellt Influence auf 0. Wenn ihr bei Animationen das Fokusobjekt mit der Kamera bewegen wollt, verwendet nicht die Kamera sondern das Trägerempty als parent für den Fokus (STRG P).



parallel



mit Fokus

Beide Bilder sind mit denselben Einstellungen gerendert ($\text{deltaX} = 0.2$), beim Rechten zielt die Kamera aber immer auf den Kopf, wodurch der Effekt etwas anders rauskommt. Ich finde es lässt sich entspannter anschauen.

Und jetzt viel Spaß mit eurer 3D-Brille und Blender. 🎮