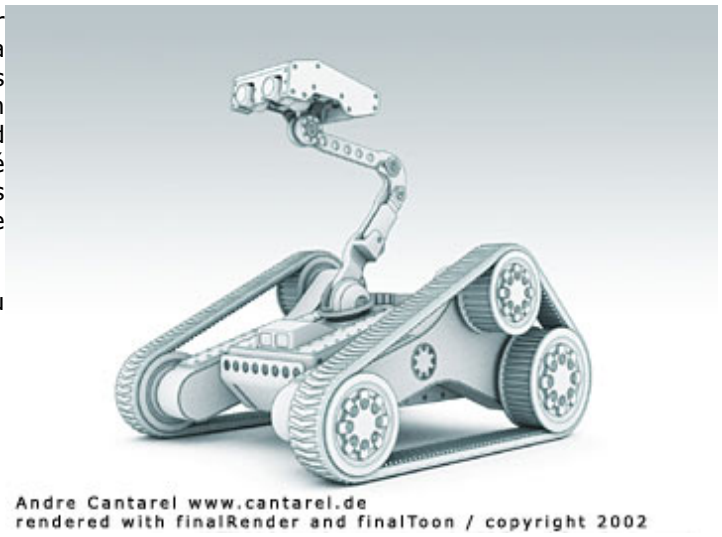
finalToon

Cebas finalToon
non official logo by Volker "Don" Uffelmann

Beaucoup d'encre à coulée sur les dates annonçées par Cebas pour finalToon, qui aurait du être sur le marché depuis bientôt trois mois. Sa commercialisation aura en fait été retardée pour permettre de finaliser les accords entre Cebas et Discreet, puisque finalToon est le premier plugin certifié par Discreet pour son nouveau programme DCP (Discreet Certified Plugin). Une garantie énorme pour les clients sur la valeur et la perennité du produit. finalToon est donc LE moteur toon officiel pour 3dsmax, Cebas en assurant le développement et Discreet la distribution, le support, le packaging, etc. Il est disponible pour 3dsmax4/5 et 3dviz4.

Assez parlé marketing, parlons un peu de ce fameux moteur de rendu cartoon !



Andre Cantarel www.cantarel.de
rendered with finalRender and finalToon / copyright 2002

- Je ne connait pas très bien les autres moteur cartoon pour 3dsmax, en dehors de quelques tests, et ne vais pas me lancer ici dans une comparaison. Quoiqu'il en soit, finalToon est de loin celui qui m'as paru dès le debut le plus simple et le intuitif.
finalToon gère bien entendu le tracé des contours, intersections, contours invisibles, etc. etc. Il permet surtout de contrôler l'aspect de ces contours, bien sur en épaisseur (qui peut varier suivant la profondeur !) mais aussi véritablement son aspect géométrique (bruit, déformation, mapping). De même, vous pouvez appliquer des motifs de remplissage (hatching) sur des surfaces, les ombres, etc. !
finalToon est également le seul moteur cartoon à gérer complètement les réflexions et réfractions ! Ceci signifie que si une ligne de contour n'est pas visible sur l'objet, mais qu'elle l'est dans la réflexion sur un autre, elle sera effectivement tracée dans cette réflexion ! Chose impossible avec d'autres moteurs, basés sur des shaders. De même, les contours invisibles dans les ombres peuvent présicement être rendus visibles ! Particulièrement appréciables dans les rendus architecturaux ou techniques. Qui plus est, finalToon se révèle particulièrement rapide, ce qui ne gâche rien.

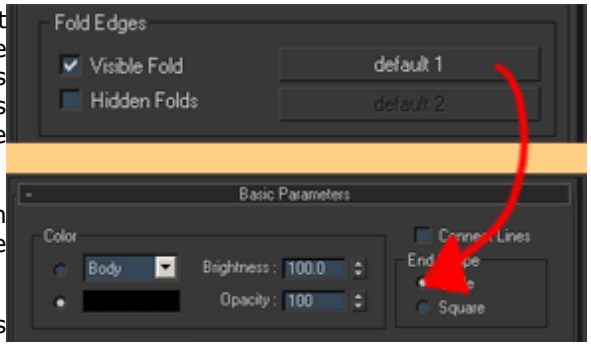
finalToon est implémenté de diverses manières : Le rendu lui même est disponible en tant que render effect, ce qui permet de l'utiliser avec n'importe quel moteur de rendu (le scanline de base ou la future Stage1 par exemple), et de visualiser les paramètres en temps réel (option Interactive des render effect). De plus, avec les options par défaut, il suffit d'activer ce render effect pour avoir un rendu illustré, sommaire certes, sans toucher aux matériaux. Vous accédez ainsi par ce render effect à tout les paramètres de contours, remplissage (hatching) etc. pour tout les objets de la scène (globals) ne possédant pas leurs propres paramètres.

- Pour un contrôle plus avancé, au niveau des matériaux, vous avez le finalToon material, avec toutes les options de shading nécessaires, et les contrôles locaux sur les surfaces et edges (les mêmes contrôles qu'on retrouve dans le render effect pour la scène dans sa globalité). A noter que vous avez à la fois un shader finalToon, et un matériau finalToon, avec lequel vous pourrez utiliser les shaders standards (phong, blinn etc) tout en contrôlant localement l'aspect des contours.

De même, finalToon gérant les réflexions et réfractions, vous avez quatre nouvelles maps : *finalToon Flat Mirror, finalToon Hatching, finalToon Reflect/Refract,* et *finalToon Thin Wall Refraction.* Cela vous permettra aussi de gérer des scènes mix entre materiaux réaliste, plastiques etc avec des matériaux cartoon, même dans les réflexions/refraxions !

finalToon ajoute également un render element, *lines* aux render elements standards de 3dsmax, de façon à séparer le tracé des contours du rendu. Vous pouvez ainsi faire une passe pour différents types de lignes, comme les contours visibles, les intersections invisbles, etc etc ! Particulièrement puissant.

■ Les contrôles sur les contours, comme je le disais plus haut, sont particulièrement poussés, et en même temps très simple à utiliser. Tout d'abords, vous pouvez faire apparaitre lors du rendu les contours visibles et/ou invisibles, de même pour les intersections. Vous pouvez également faire apparaitres des contours entre des IDs différentes de matériaux, et aussi utiliser les fonctions visible edge/invisible edge d'un edit mesh pour decider quelles edges d'un mesh seront déssinées lors du rendu.



A gauche, vous avez les deux interfaces liées au contrôle des contours (cliquez en haut ou en bas de l'image). Ces paramètres sont interchangeable, copiable etc entre chaque type de edges (visible/invisble intersection etc).

Vous voyez ainsi les contrôles applicables aux contours. Notez également les possibilités de mapper les contours, particulièrement efficace pour ajouter du grain, des irrégularitées, etc !
Vous apercevez aussi les options d'antialising appliqué aux contours, qui evite d'obtenir du moirage par exemple.

■ Le hatching : le hatching prend la forme d'une map (*finalToon Hatching*) et permet d'ajouter des sortes de coups de crayon dans certaines zones du rendu. Cette map permet de prendre un bitmap (par exemple, un coup de crayon) et de le répéter, avec des variations bien sur, dans différents canaux de votre matériau. Par exemple dans les ombres, ou seulement dans les highlights, etc etc. Vous pouvez ainsi vous rapprocher assez facilement d'un rendu crayonné, avec cette map plus que quelques variation en épaisseur/opacité dans les contours.


finalToon Hatching
image by Thomas Pardun

Dans le même registre, vous pouvez également spécifier un fond particulier simulant un dessin sur une feuille spécial. Per exemple un papier granuleux assez épais. finalToon prendra en compte l'aspect granuleux pour le "dessin" du rendu sur une telle feuille.

■ Export Flash et AI

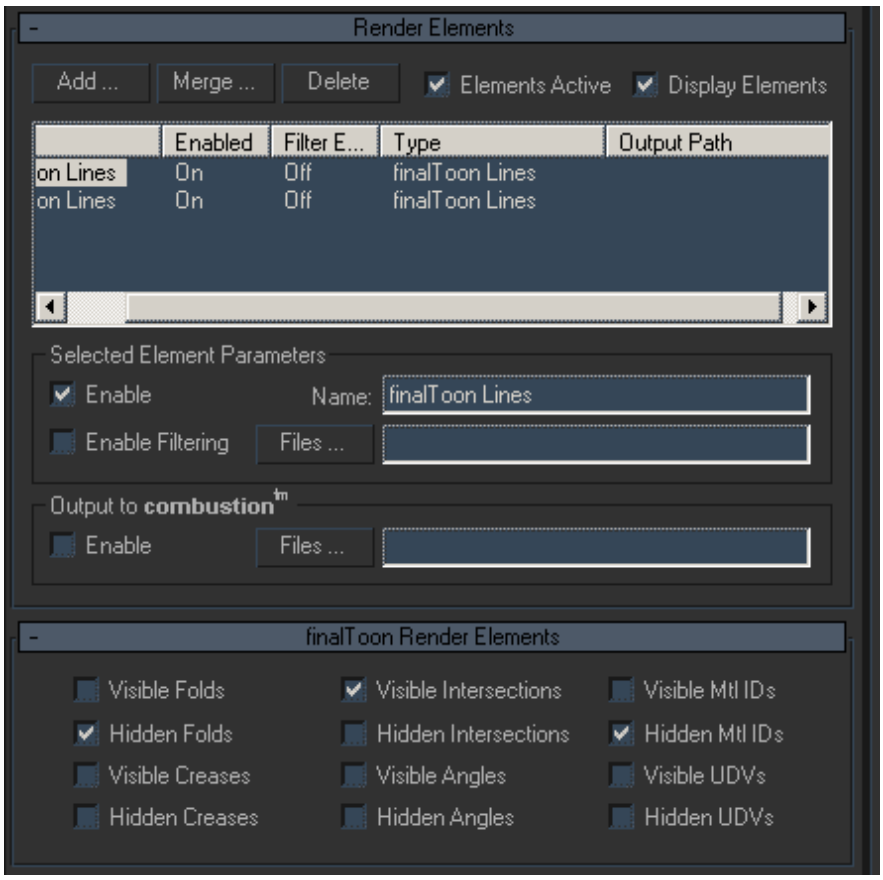
finalToon permet d'exporter le rendu en .ai (format illustrator), qui est un des standards du format vectoriel, lu par la plupart des logiciels vectoriels. Il supporte également la création de plusieurs calques séparés (avec Illustrator 10).

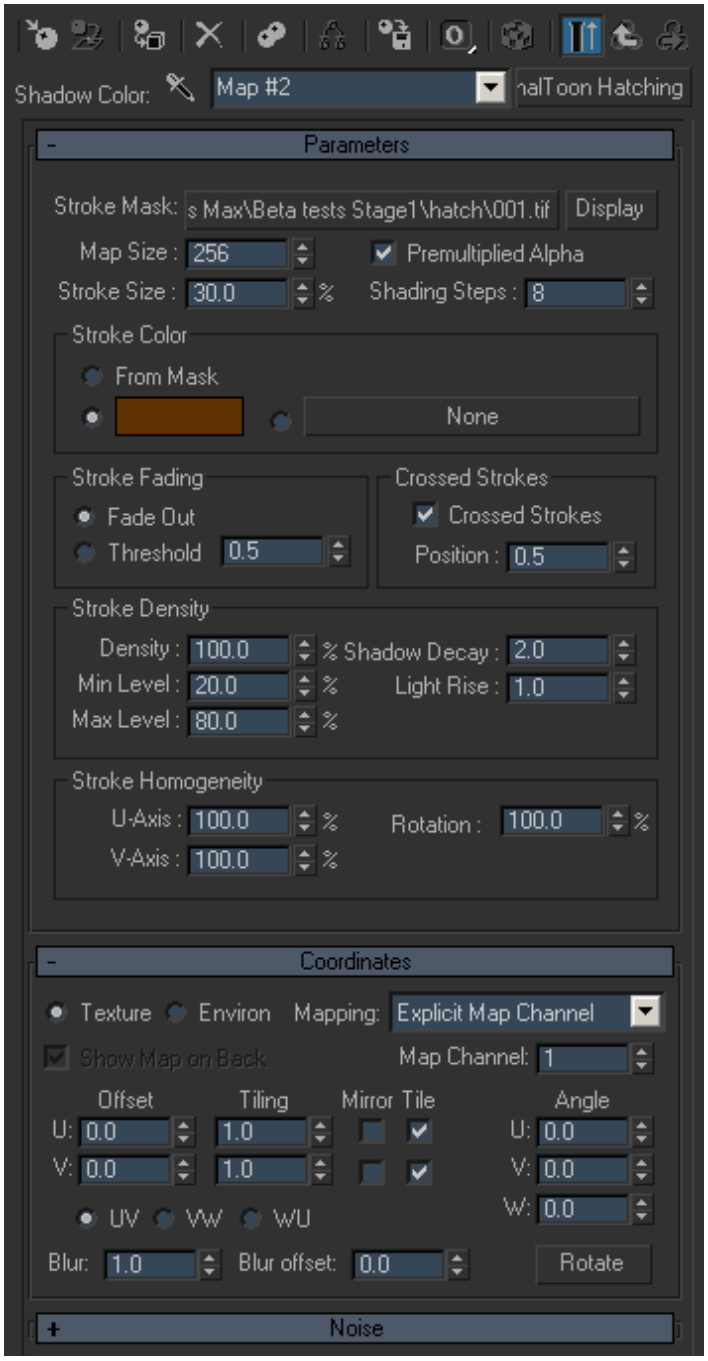
finalToon permet également d'enregistrer les rendus et animations au format Flash. Un seul regret toutefois, et de taille, l'exporter flash n'exporte que les contours. Cela devrait être réglé prochainement, une solution consistant dès maintenant à faire l'export en .ai, que flash importe également. Valable pour les images fixes, mais pas vraiment pour les animations :-(Update à venir...

■ Je n'ai bien évidemment pas tout décrit en détail, mais j'espère que je vous aurai aider à comprendre les possibilités et l'utilisation de ce fantastique moteur de rendu. Je n'ai moi même pas participer très activement aux alpha/beta tests, mais à chaque fois que j'y ai touché, je l'ai toujours trouvé très simple d'utilisation et très intuitif. Ce qui, avec un potentiel comme celui-là, augure de futurs nombreux et beaux rendus NPR (Non Photorealistic Render) ! Je rappel que finalToon sera inclut avec la Stage1, sans supplément de prix !

Jetez aussi un oeil sur le site de PixelGrind, qui a crée des animations avec fT, et également sur Destiny, par Steve Weber.

Nicolas Genette, 20/01/2003


Render Elements
Add ...
Merge ...
Delete
Elements Active
Display Elements
Enabled
Filter E...
Type
Output Path
on Lines
On
Off
finalToon Lines
on Lines
On
Off
finalToon Lines
Selected Element Parameters
Enable
Name:
finalToon Lines
Enable Filtering
Files ...
Output to combustion
Enable
Files ...
finalToon Render Elements
Visible Folds
Visible Intersections
Visible Mtl IDs
Hidden Folds
Hidden Intersections
Hidden Mtl IDs
Visible Creases
Visible Angles
Visible UDVs
Hidden Creases
Hidden Angles
Hidden UDVs


Shadow Color:
Map #2
nalToon Hatching
Parameters
Stroke Mask:
s Max\Beta tests Stage1\hatch\001.tif
Display
Map Size :
256
Premultiplied Alpha
Stroke Size :
30.0
%
Shading Steps :
8
Stroke Color
From Mask
None
Stroke Fading
Fade Out
Threshold
0.5
Crossed Strokes
Crossed Strokes
Position :
0.5
Stroke Density
Density :
100.0
%
Shadow Decay :
2.0
Min Level :
20.0
%
Light Rise :
1.0
Max Level :
80.0
%
Stroke Homogeneity
U-Axis :
100.0
%
Rotation :
100.0
%
V-Axis :
100.0
%
Coordinates
Texture
Environ
Mapping:
Explicit Map Channel
Show Map on Back
Map Channel:
1
Offset
Tiling
Mirror Tile
Angle
U:
0.0
1.0
U:
0.0
V:
0.0
1.0
V:
0.0
W:
0.0
UV
VW
WU
Blur:
1.0
Blur offset:
0.0
Rotate
Noise