

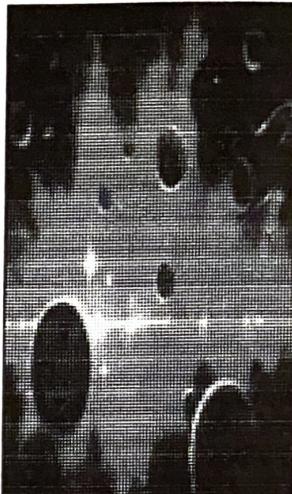
EDITO

Vous avez sûrement remarqué le changement dans la présentation de la newsletter : celle-ci a changé de logo. Oui mais quel logo ? Ce sera à chacun de choisir entre les deux qui sont présentés au recto et au verso de ce numéro. L'un est en 3 dimensions et a été tracé avec POV, l'autre est en 2 dimensions et a été réalisé avec DeluxePaint. Les réponses seront attendues jusqu'à fin Juillet.

Les bicubic patches, plus particulièrement les surfaces bázier, sont le sujet de ce mois. Les questions/réponses continuent leur chemin dorénavant habituel, et enfin une nouveauté assez surprenante : le port de Fractint sur Amiga.

Si malgré le soleil de Juillet vous avez l'occasion d'écrire quelques petits articles vous pouvez les envoyer. La newsletter du mois d'Aout dépendra de votre participation à tous.

- Nicolas Mougel



Planètes

LES BICUBICS PATCHES

La représentation en 3 dimensions des surfaces s'effectue généralement par le biais de deux méthodes : les *polygon meshes*, ou bien les *bicubic patches*.

Un *polygon mesh* est une série de surfaces planes polygonales connectées les unes aux autres. Cette technique a comme inconvénient de donner une représentation approximative d'un objet puisque la fidélité de la représentation d'une forme dépend du nombre de points qui sont utilisés, et donc du nombre de *polygon meshes* nécessaires. Si l'on veut une très bonne approximation d'une forme, cela coûte cher en mémoire, et en temps machine.

Un *parametric bicubic patch* permet de définir les coordonnées de tous les points d'une surface donnée grâce à l'utilisation d'équations. On utilise donc le paramétrage pour définir la surface, et à chaque coordonnée x, y et z, on associe une équation, avec 2 paramètres ayant comme puissance maximale 3 (d'où les termes bi et cubic). Pour créer une surface complexe, il est nécessaire de joindre plusieurs bicubic

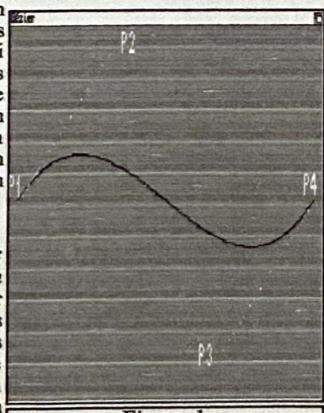


Figure 1

patches les uns aux autres. L'inconvénient de cette méthode vient du fait que les algorithmes qui traitent des bicubic patches sont plus longs que ceux nécessaires aux *polygon meshes*.

Nous allons nous intéresser aux bicubic patches et plus particulièrement aux surfaces de Bézier. Mais tout d'abord traitons des courbes de Béziers.

COURBES DE BEZIER

Abordons rapidement la partie mathématique. Une courbe cubique paramétrée se définit de la manière suivante :

$$\begin{aligned}x(t) &= axt^3 + bxt^2 + cxt + dx \\y(t) &= ayt^3 + byt^2 + cyt + dy \\z(t) &= azt^3 + bzr^2 + czt + dz\end{aligned}\text{ où } 0 \leq t \leq 1$$

Remarque : ax signifie a indice x, et t^3, t puissance 3.

En général, la représentation d'une courbe dans l'espace est assez dure à représenter sur un écran, donc on s'intéresse juste au plan dans notre cas.

Abordons maintenant le vif du sujet : Une courbe de Bézier est définie par 4 points de contrôle p1, p2, p3, p4. Le point p1 représente le début de la courbe, p4 sa fin. Les points p2 et p3 permettent quant à eux de contrôler l'aspect de la courbe, et ainsi d'offrir à un utilisateur la possibilité de choisir une courbe telle qu'il la désire en modifiant seulement les coordonnées de p2 et p3. Pour ceux qui veulent en savoir plus, les vecteurs tangents à la courbes sont déterminés grâce aux segments p1p2 et p3p4.

Pour voir comment agissent les différents points de

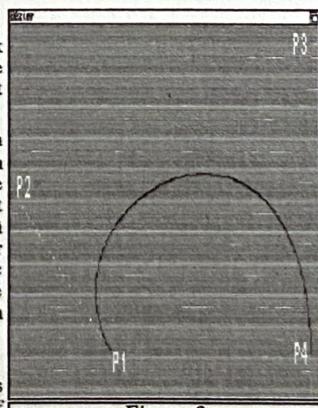


Figure 2

contrôle sur la courbe de Bézier reportez-vous aux figures 1, 2, 3 et 4.

Pour chaque figure, voici les coordonnées qui ont été utilisées :

	p1x	p1y	p2x	p2y	p3x	p3y	p4x	p4y
Figure 1	0	0	1	1	2	-1	3	0
Figure 2	1	0	0	1	3	2	3	0
Figure 3	5	0	0	1	15	1	10	0
Figure 4	0	0	5	0	10	0	12	5

Il est à remarquer que les courbes Bézier n'ont pas la même échelle en x qu'en y ; Les échelles dépendent des valeurs utilisées.

Comme vous pouvez le voir, les courbes obtenues sont assez diversifiées. C'est à partir de ces courbes, que les surfaces de Bézier vont être construites.

SURFACES DE BEZIER

Elles sont donc une extension des courbes de Bézier.

La définition mathématique de la surface est la suivante :

$$x(s,t) = a11s^3t^3 + a12s^3t^2 + a13s^3t + a14s^3 + a21s^2t^3 + a22s^2t^2 + a23s^2t + a24s^2 + a31st^3 + a32st^2 + a33st + a34s + a41t^3 + a42t^2 + a43t + a43$$

Idem pour y(s,t) et z(s,t).

Remarque : a11 signifie a indice 11.

J'en vois déjà qui s'effraie. Rien à craindre, nous ne nous utiliserons pas cette formule dans les explications qui vont suivre, celle-ci étant donnée à titre d'information.

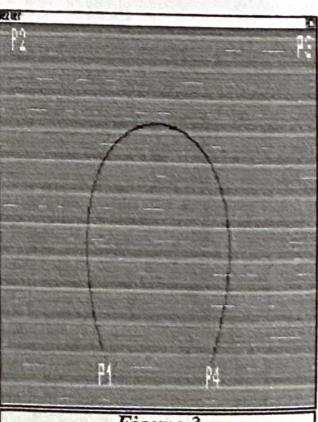
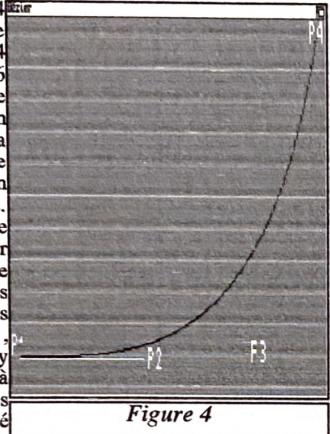


Figure 3

Vous vous rappelez qu'une courbe de Bézier est définie par 4 points de contrôle. Comme on s'intéresse dorénavant aux surfaces, ces 4 points de contrôle deviennent 4×4 , c'est à dire 16 points de contrôle. Jetez un coup d'œil à la figure 5 pour que mon explication soit convaincante. Remarque : Le tracé en noir sur la figure ne présente que les courbes extérieures, comprenez qu'il y en a aussi à l'intérieur. Elles n'ont pas été représentées pour éviter d'allonger le dessin.


Figure 4

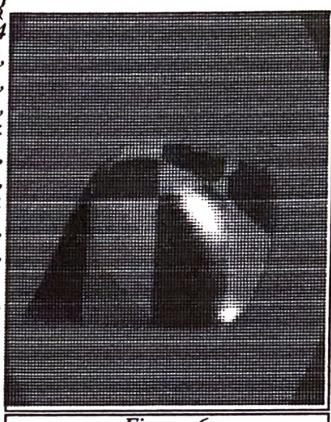
Ainsi la surface de Bézier est définie par 16 points de contrôle. C'est les fameux 16 paramètres nécessaires lorsqu'on utilise un bicubic patch dans le raytracer POV.

Pas de miracle : les points de contrôle ne peuvent pas se trouver facilement à la main. En général on utilise un logiciel qui permet de construire une surface de Bézier à partir de points de contrôle. Comme on peut voir le résultat à l'écran, il est alors facile de créer la surface voulue. Lorsqu'aucun logiciel de ce type n'est disponible, une alternative s'offre à vous : A partir de courbes de Bézier que vous connaissez (c'est à dire les points de contrôle et la forme obtenue), vous extrapolez les résultats sur la surface. Avec un peu de pratique, c'est jouable ; c'est sûr que ce n'est pas la solution rêvée.

Pour terminer, voici une courbe de Bézier, en figure 6 donc, inspirée du livre RayTracing Creations, dont le

rendu a été réalisé avec POV. Les points de contrôle sont les suivants :

```
bicubic_patch {
    type 1 u_steps 4
    v_steps 4 < 0,
    0, 0>, < 1, 0,
    0>, < 2, 0, 0>,
    < 3, 0, 0>, <
    0, 0, 1>, < 1, 3,
    1>, < 2, 3, 1>,
    < 4, 0, 1>, <
    0, 0, 2>, < 1, 2,
    2>, < 2, 1, 2>,
    < 4, 2, 2>, <
    0, 0, 3>, < 1, 0,
    3>, < 2, 0, 3>,
    < 3, 0, 3> }
```


Figure 6

Dernière minute : Sur les figures 1 à 4 les traits en bleu sont des tangentes à la courbe ne sont pas très visibles. Par exemple pour la première figure, deux tangentes sont présentes : p1p2 et p3p4.

- Nicolas

QUESTIONS/REPONSES

Réponse de Nicolas pour Wai Yip :

Les verres (et non le verre) constituent un ensemble extrêmement varié de produits dont les propriétés sont innombrables. Exemple : Jusqu'en 1880, le verre courant est appelé crown ordinaire et a pour indice de réfraction 1.51. Pour obtenir un verre plus brillant, une composition nouvelle a été mise au point, et on a obtenu le flint dont l'indice de réfraction est de 1.71.

Pour POV, il faut utiliser les déclarations Crown Glass Ior et Flint Glass Ior qui se trouvent dans le fichier ior.inc.

Pour les courbes de Béziers, lis l'article de ce mois. Sinon en ce qui concerne les paramètres u_steps et v_steps, ceux-ci servent à déterminer le nombre de triangles qui vont représenter la courbe. Ex : u_steps = 5 et v_steps = 5 entraînent 5x5 = 25 triangles.

Réponse de Charles pour Arnaud :

Je joins quelques pages d'assembleur à ton intention concernant le passage RGB/HSL. Ce sont ces routines que j'utilise dans mes palettes.

NEWS

FRACTINT

FRACTINT est à la base un logiciel de fractals créé en 1988 sur PC et écrit par Bert Tyler. Depuis il s'est imposé comme le logiciel de fractals sur PC, surtout aux Etats-Unis où des images et formules sont échangées sur Réseaux.

Le port Amiga vient probablement du port Unix, puisque celui-ci est indiqué à l'introduction du programme. Pour utiliser FRACTINT il vous faudra un Amiga 1200 équipé soit de beaucoup de mémoire, soit d'un disque dur, plus de la mémoire

fast. En effet, l'utilisation sur disquette n'est pas envisageable. Une version pour ceux qui possède un coprocesseur est disponible.

Venons maintenant au logiciel. La fenêtre de contrôle a gardé son aspect PC et tout se fait au clavier, la souris n'intervenant que lors des zooms sur les fractals. On peut plutot parler d'une beta-version pour le port Amiga puisque celui-ci présente des faiblesses par rapport à la version originale. Ainsi les arguments que l'on peut entrer sous DOS après la commande fractint ne sont pas supportés sur Amiga, le logiciel indiquant à chaque fois le message unixarg. L'interface de la fenêtre de contrôle aurait pu être repensée. En effet celle-ci sera juste à l'affichage, la souris ne s'utilise pratiquement

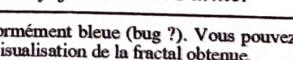
jamais et aucun gadget n'est pris en compte (même pas le gadget de fermeture de la fenêtre). L'utilisation du logiciel en souffre donc beaucoup et n'est pas user-friendly du tout. Autre point noir : Les couleurs des palettes proposées ne sont pas harmonieuses et un effort est à fournir de ce côté. De même le changement des couleurs par le biais d'un espace de requête qui apparaît à l'écran est à déconseiller tellement c'est à incompréhensible. L'ergonomie est donc largement à revoir.

Les points forts maintenant. L'argument le plus important de fractint est la diversité des fractals proposés : Mandelbrot, Julia, Newton, Plasma, L-Systems, Lyapunov, les Quaternions, etc... Le plasma est vraiment à voir. En plus de ce grand choix de fractals, fractint vous offre la possibilité de créer de nouvelles fractals de votre cru en indiquant la formule que vous désirez utiliser dans un fichier dédié. Des formules supplémentaires dans d'autres fichiers sont présentes (avec des palettes acceptables cette fois) et sont intéressantes à découvrir. Ce sont ces types de fichiers qui sont échangés sur réseaux apparemment. L'aspect des fractals est aussi très paramétrable. Ainsi on peut choisir la méthode de tracé (par les frontières, en

1 passe, en 2 passes, etc...), le type de représentation par couleur (le biomorph par exemple a un très beau rendu), ainsi que d'autres paramètres spécifiques à chaque fractal. De ce côté fractint est bien fourni, et je ne vais pas trop m'étendre dessus car il faut avoir le logiciel sous la main pour comprendre tous les choix qui sont proposés.

Enfin, une autre possibilité est celle de transformer les images fractals dans le plan en fractals dans l'espace en affectant une hauteur à chaque couleur. On peut obtenir ainsi des objets à utiliser pour DKB Trace, Vivid, RayShade, etc... Dommage que pour POV l'implémentation ne soit pas faite. Si vous désirez sauvegarder l'image directement c'est normalement possible, le seul problème de taille sur Amiga c'est que l'image obtenue au format tga n'est pas conforme à ce qu'on doit en attendre, puisque celle-ci est uniformément bleue (bug ?). Vous pouvez cependant avoir une prévisualisation de la fractal obtenue.

En conclusion, Fractint doit dépasser le stade de beta version pour être vraiment reconnu comme un bon logiciel de fractals sur Amiga, même si celui-ci renferme un potentiel énorme de création.


Nyuf004 - Dan Farmer

Nyuf004 - Dan Farmer

Merci à François pour le logo Wild Graphics en 2D ainsi que Waihip pour celui en 3D. Merci également à Charles Vassalo pour sa participation au niveau des questions/réponses.

Pour tout envoi d'articles :
Wild Graphics, 6 avenue de la Chasse, 77500 Chelles