

*W*ILD *G*RAPHICS *N*EWSLetter

Numéro 13

La newsletter des passionnés de l'image

Décembre-Janvier 1996



Interview : David K. Buck

Techno-Logique : Les B-Splines

La Presse : >InterActif, Wired

News : POV v3.0, ToyStory, LParser

copyright 1995 Robert R. Nielsen

ABSOLUT EDITO

nicolas.mougel@etu.univ-compiegne.fr

News



de notre envoyée spéciale...

Toy Story

Dans le numéro 11 de la newsletter, il était question de la sortie du dessin animé réalisé entièrement en images de synthèse, le nommé **Toy Story**. La sortie s'est faite en Novembre 1995 aux Etats-Unis. Apparemment il plait aussi bien aux petits qu'aux grands là-bas. Vivement donc sa sortie en France. Elle devrait se faire en Mars de cette année. Je comptais vous parler un peu plus de l'histoire que j'ai pu lire sur un site web, mais voilà, j'ai perdu mes notes...

Persistence Of Vision version 3.0

La version 3.0 de POV est en cours de préparation... Au menu, pas mal de nouveautés comme la possibilité de créer des arcs-en-ciel, ainsi qu'un langage script amélioré. Sur toute plateforme.

Pour la petite histoire, POV est originaire de l'Amiga, eh! oui... Vous en apprendrez plus en lisant l'interview de ce numéro. L'Amiga est vraiment resté une machine

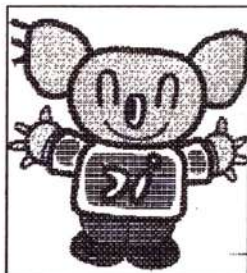
conviviale... Je n'ai trouvé aucune plateforme où il y ait des éditeurs de textes aussi agréables à utiliser que sur Amiga...

LParser

Une nouvelle version du logiciel PC de L.J. Lapre nommé L-System Parser vient de sortir. Rappelons que LParser permet entre autre de créer des plantes et des insectes, ceci en utilisant un langage basé sur le système de LindenMayer. Au finish, on obtient des objets assez insolites qui peuvent être réutilisés dans des logiciels tels que 3D Studio. Téléchargeable sur <http://www.xs4all.nl/~ljlapre>

Presse

>InterActif



7ème numéro pour ce récent magazine nommé InterActif centré nouvelles technologies donc pas mal axé InterNet et technologie numérique. C'est un magazine bimestriel disponible au prix de 30 F. Pourrais-t'on parler d'un Wired français ? Sans doute. Il est vrai que les éditos ils ne savent pas les faire (quand on a pas d'inspiration le plus simple c'est de ne pas en faire n'est-ce pas ?). Toutefois on y trouve des articles intéressants, pas mal dans la lignée de la cyberculture. Vous connaissez sans doute de nom William Gibson ? Vous savez il a écrit Cablé. Y paraît que c'est une référence pour la cyberculture (Désolé mais la version française est bourrée de fautes et pas vraiment réussie, c'est plutôt au niveau des idées qu'il y avait une révolution mais

la traduction est vraiment limitée à part la première phrase qui à elle seule vaut peut être le coût d'acheter le livre). Enfin parenthèse mise à part, si je parlais un peu du contenu de ce numéro. Eh bien, il y a une rubrique de news appelé Mix. On y trouve notamment une brève sur le safe-sex, ou comment découper un carré dans un gant en latex pour tout rapport bucco-vaginal (c'est sur Internet... il fallait s'en douter). Voici quelques articles parus dans ce numéro... Quand on apprend que les robots nous piquent nos emplois, que dieu est joignable sur Internet, que sont les BarbiTwins, que tonton est peut-être joignable en écoutant votre neige cathodique, que les agents sont intelligents (pas les agents de police), qu'il y a une cyber-littérature, que vous pouvez créer votre page Web... et qu'il y a des bons plans sur le net. Oui, à ce propos si vous désirez connaître l'adresse électronique de votre copain roger mais comme par hasard celui-ci a oublié de vous la donner, c'est simple vous pouvez la retrouver en vous connectant sur : <http://www.iaf.net/> La recherche se fait par nom, voire par prénom un peu comme sur le 11 du minitel (il commence à se faire vieux celui-là d'ailleurs). En bref, >InterActif c'est un magazine à conseiller

WIRED

Manteau blanc pour la couverture du dernier numéro de Wired. A l'intérieur des pages en gris-métallisé et blanc vous emmènent au fil de votre lecture : le futur du shopping électronique, dossier sur Marshall McLuhan, qui sommes-nous, la propriété intellectuelle, comment écrire un nouveau scénario pour la série Star Wars... Vous apprendrez également dans le 'Colophon' en fin de numéro tout sur l'équipe de Wired. Comment est réalisé le magazine ? Quel est le hardware utilisé (Macintosh) ? Le software nécessaire (QuarkXPress) ainsi que le bearware ! Pas

mal de publicités dans ce numéro somme toute, et le dossier est peu intéressant alors qu'en général c'est ce qui fait la force de chaque numéro. Dommage. Dans toutes les bonnes charcuteries anglophones.

interview



David K. Buck

L'e-mail réduit les distances à une vitesse vertigineuse... Lors de l'interview précédente, il était question du groupe français ChromaGraphics, cette fois direction le Canada où David K. Buck a bien voulu m'accorder une interview e-mail...

> Pouvez-vous vous présenter ?

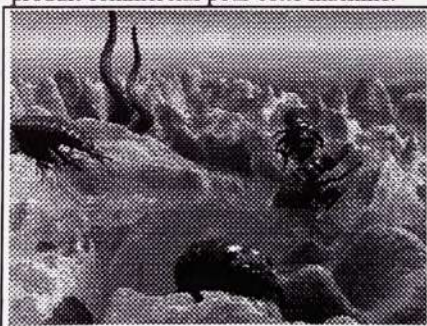
Je m'appelle David Kirk Buck. Je suis né et j'ai grandi au Canada où je vis actuellement. J'ai toujours été intéressé par la programmation mais mon intérêt a vraiment débuté en 1978 à l'âge de 15 ans. A l'époque, j'ai construit mon premier ordinateur (basé sur le microprocesseur 1802). Plus tard je suis passé au TRS-80 Couleur, puis à l'Amiga.

" Un jour de 1985..."

> C'est vous qui avez développé le logiciel DKB Trace en 1989 (si mes souvenirs sont exactes). Il s'agissait d'un logiciel de raytracing très connu. Qu'est ce qui vous a poussé à le faire à l'époque ?

J'ai toujours eu un intérêt pour le domaine des images de synthèse, et j'ai été époustoufflé par le film Tron et les quelques publicités télévisées de l'époque

qui étaient faites par ordinateur. Un jour de 1985, un de mes amis m'a donné un programme de raytracing qu'il avait trouvé sur Internet. Il était écrit en C et fonctionnait sur système UNIX. Seuls les niveaux de gris étaient possibles et les objets que l'on pouvait utiliser se limitaient aux sphères sur sol fixe de type échiquier. Aucun de nous deux avait accès à une machine UNIX pour lancer le programme, j'ai alors décidé de le porter sur Amiga en utilisant le compilateur Manx Aztec C. J'ai continué à travailler dessus pour ajouter la couleur et des textures en utilisant les techniques décrites dans l'article de Ken Perlin "Image Synthesis" paru dans le SIGGRAPH de 1984. Voilà. La bête du raytracing m'avait piqué. A partir de là, j'ai voulu en faire plus avec le programme mais la structure du programme que j'avais était bien pauvre et il était difficile de l'étendre. J'ai décidé de réécrire tout depuis le début en utilisant des techniques apprises en utilisant le SmallTalk durant mon programme de licence. Le nouveau programme -DKBTrace - était un raytracer orienté objet et écrit en C. Rapidement, je m'étais constitué un raytracer valable. La question était ensuite qu'est ce que j'allais en faire ? Etant donné que tout le logiciel était écrit pour l'Amiga j'ai pensé créer un produit commercial pour cette machine.



Bug - Mike Miller

Mais il y avait diverses difficultés liées à cela. En ce temps (vers 1987) il y avait déjà différents programmes d'image de synthèse disponible pour l'Amiga tels que VideoScape ou TurboSilver et tous comprenaient une interface graphique. Je n'avais pas le temps ou l'énergie d'écrire

une bonne interface graphique étant donné que je venais juste de commencer mes études de maîtrise. J'ai aussi pensé que sans interface graphique cela ne ferait pas un très bon produit commercial et je ne voulais pas le mettre en shareware parce que je ne voulais pas maintenir le logiciel sachant que je n'en avais pas le temps. Je me suis beaucoup amusé en le développant alors je l'ai lancé en tant que freeware pour que d'autres puissent comme moi s'amuser avec. DKBTrace 1.0 était né suivi après par les versions 1.1 et 1.2 avec de nouvelles fonctionnalités et des corrections de bogues.

> Pensiez-vous au départ porter votre programme sur d'autres plateformes telles que le PC ? C'est d'ailleurs Aaron Collins qui s'en est chargé non ?

Vers 1988 si ma mémoire est exacte, j'ai reçu un appel de la part de Aaron Collins qui vivait à Chicago. Il avait vu mon programme DKBTrace sur un BBS local et il avait décidé de le porter sur PC afin qu'il puisse fonctionner dessus. Il avait continué à ajouter de nouvelles fonctionnalités comme l'éclairage de Phong et quelques nouvelles textures. Il m'appelait pour savoir si j'étais intéressé par ces changements et pour les inclure dans la prochaine version de DKBTrace. Pour être honnête, je ne comptais pas sortir une nouvelle version de DKBTrace mais je lui ai demandé de m'envoyer les changements de manière à ce que je puisse les voir. Il m'a envoyé le nouveau code-source mais je ne pouvais pas facilement prendre le code pour le compiler et le lancer sur mon Amiga. Après avoir finalement réussi à le faire, je fus très impressionné par les nouvelles possibilités et j'ai demandé à Aaron s'il était intéressé pour travailler avec moi dans le développement d'une nouvelle mise à jour incluant ces nouveautés et quelques autres. Il a accepté. Nous avons décidé que si nous voulions continuer dans cette voie là, nous devions faire en sorte que DKBTrace soit portable. Nous avons passé plusieurs mois

à nous envoyer des mises à jour à l'un et l'autre et faire le programme aussi portable que possible. Je l'ai essayé sur stations Sun à l'université de Carleton et j'ai pu le lancer dessus également. Après au moins un an de travail et quelques centaines de dollars utilisés pour les notes téléphoniques longue distance nous avons créé DKBTrace version 2.0 et l'avons lancé. Dans l'univers de l'Amiga, les raytracers étaient très communs mais sur PC, ils étaient quasiment inconnus et DKBTrace s'est propagé comme un feu sauvage. Des gens commencèrent à nous envoyer des images fabuleuses et du code pour de nouvelles fonctionnalités. Nous avons commencé à être vite dépassés par les réponses. En fait, je trouvais ça marrant. C'était intéressant de voir des gens vraiment passionnés par un programme que vous aviez écrit et qui envoyaient des images faites avec ce logiciel. Il y eut aussi beaucoup de demandes pour ajouter de nouvelles fonctionnalités. Pendant un moment, Aaron et moi avons ajouté de nouvelles choses à DKBTrace - après tout ça nous avait passionné aussi. Mais nous avons commencé à trouver que DKBTrace nous prenait beaucoup de temps. Nous devions arrêter à un moment ou à un autre mais nous n'aimions pas l'idée de tuer le programme alors que tant de gens l'appréciaient. C'est à cette époque en 1991 que j'ai eu un compte CompuServe parce que j'avais entendu dire qu'il y avait beaucoup de gens sur CompuServe qui étaient passionnés par DKBTrace. Après avoir lu beaucoup de messages de leur part, il était clair qu'ils désiraient avoir de nouvelles fonctionnalités ajoutées à DKBTrace mais Aaron et moi étions trop lents à ajouter ces fonctionnalités au programme. Il était alors question pour ces personnes d'écrire un nouveau raytracer depuis le début (pour éviter des problèmes de propriété) et d'ajouter les nouveautés qu'ils voulaient à ce nouveau raytracer. Aaron et moi avons vu une solution au problème. En mars 1991, j'ai posté un message sur CompuServe avec l'offre

suivante : les gens sur ce forum pourraient constituer une équipe pour créer un raytracer en utilisant DKBTrace comme une base. Les seules conditions étaient : 1) Utiliser un nom différent - Je ne voulais pas que mes initiales soient liées au programme pour toujours 2) Le programme reste librement distribuable 3) Le programme reste aussi portable que possible. Ils aimèrent l'idée. Nous avons décidé que le tournant se ferait après la dernière version de DKBTrace - la version 2.12. Cela s'est produit en Août 1991.



Tonka - Mike Miller

" comme un feu sauvage... "

> DKB Trace est donc ensuite devenu POV repris alors par Drew Wells en 1991. Comment s'est faite la transition et pourquoi avoir arrêté dans le domaine ? Vous pensiez être satisfait intellectuellement ?

Après que le projet POV-Ray ait commencé, je ne m'en suis pas vraiment écarté. En réalité, la première année et demie, je pris une grande part dans la contribution de POV-Ray. Après cela, j'ai commencé à travailler sur un nouveau produit conçu pour les ventes commerciales et je n'avais plus trop le temps pour POV-Ray. Je savais que le programme était en bonne main et j'étais content de laisser l'équipe continuer son développement.

> Est-il vrai que le logiciel a été fait au départ sur l'Amiga ? L'Amiga

était-elle votre machine de prédilection ? Et aujourd'hui qu'en est-il ?

DKBTrace a bien été développé au départ sur mon Amiga 1000 et plus tard sur mon Amiga 2500. C'est difficile de dire si l'Amiga était ma machine favorite. J'ai possédé 8 ordinateurs depuis que j'ai commencé en 1978, et chacun d'eux avait des qualités que j'appréciais vraiment beaucoup et d'autres qui me passaient au dessus de la tête. J'ai vraiment apprécié l'Amiga pour son architecture graphique. L'idée de programmer de belles applications graphiques sur l'Amiga m'a toujours passionné. L'environnement AmigaDOS n'a pas attiré mon attention, mais il était bien bien mieux que MS-DOS. Aujourd'hui je possède encore mon Amiga mais je m'en sers essentiellement en tant que FAX. Je suis actuellement dans le développement d'un programme pour des ventes commerciales et pour autant que je puisse aimer l'Amiga, le marché pour les logiciels Amiga n'est pas assez large. En ce moment, j'utilise la plupart du temps un Pentium 75 MHz sous Windows '95. Je ne me suis jamais trop intéressé aux PCs et je n'aime pas MS-DOS, mais Windows reste très tolérable. Pour moi, l'Amiga était révolutionnaire à son époque mais il a été abandonné il y a quelques années et n'a jamais pu combler son retard avec les systèmes PC et Mac. Aujourd'hui l'Amiga commence à être de l'histoire ancienne. C'est un comble car lorsqu'il a été lancé sur le marché il était bien au delà de tout.

> Que pensez-vous de POV aujourd'hui ?

Je pense que l'équipe de POV-Ray a fait un incroyable travail dans ce programme. Il est devenu le raytracer freeware le plus populaire et disponible sur toute plateforme. Je suis extrêmement fier d'avoir joué une part dans son développement et même si je n'y participe plus, je me sens un parent proche pour le programme et cette équipe. Le dévouement et l'énergie des membres n'a

jamais cessé de m'étonner.

> Avez-vous des sources de documentation qui vous paraissent primordiales pour savoir faire un logiciel de raytracing ?

Il y a quelques bons livres de The Waite Group sur le raytracing avec POV-Ray (NdN : "Ray Tracing Creations"- By Drew Wells - Waite Group Press). C'est un bon moyen de commencer. Si vous voulez vraiment écrire un ray tracer, il faut lire "Image Synthesis" de Ken Perlin venant de SIGGRAPH '84 conference proceedings. Les notes de cours pour les cours du SIGGRAPH sur le raytracing et les textures procédurales sont aussi de très bons documents. La série "Graphics Gems" est aussi une bonne source de documentation si vous écrivez votre propre raytracer. Enfin, il y a également un excellent livre de Andrew Glassner appelé "An introduction to Ray Tracing" (NdN : Andrew S. Glassner - Academic Press 1989) que j'ai souvent utilisé en développant POV-Ray.

> Pourquoi avoir choisi de diffuser les sources C de DKB Trace ? Est-ce que la célèbre collection Fred Fish vous a influencé dans votre choix ?

La collection Fred Fish sur l'Amiga a certainement influencé ma décision de choisir que DKB Trace soit freeware. C'était comme un frisson d'avoir son programme inclu dans la célèbre collection Fred Fish. J'admirais le travail qu'il faisait et ses efforts ont eu une grande influence sur la communauté Amiga. Avec DKB Trace mon intention était de permettre à autant de personnes que possible de pouvoir s'amuser avec le programme. Si Fred Fish n'avait pas eu cette collection, ça aurait été beaucoup plus dur de le faire.

> Admirez-vous des artistes tels que Mike Miller ou d'autres dans le domaine de l'image de synthèse ?

Une des plus grandes sensations pour moi

fut de voir des artistes comme Mike Miller et Truman Brown utiliser DKBTrace et plus tard POV-Ray pour créer de véritables oeuvres d'art. Lorsque je créais des scènes avec DKBTrace c'était pour tester certaines fonctionnalités du programme. Les images résultantes pouvaient être impressionnantes techniquement mais elles n'étaient pas stupéfiantes. Les images de ces artistes l'étaient.



Dragon - Mike Miller

" Le haut réalisme devient ennuyeux. Retournons au fun "

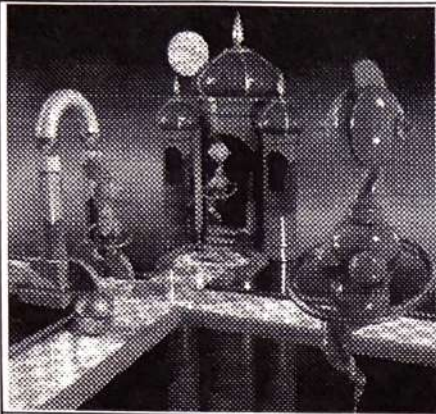
> Comment situez-vous aujourd'hui le domaine de l'image de synthèse dans la vie de tous les jours ? (de plus en plus présent ?)

Il est vrai que les images de synthèse sont de plus en plus utilisées dans les films, les publicités, et les jeux sur ordinateur. La tendance est vers de plus en plus de réalisme, de meilleures textures, de meilleurs éclairages et d'effets spéciaux. Je trouve cependant qu'après avoir travaillé dans le graphisme de haute-qualité pendant 10 ans, sa nouveauté s'est passée chez moi. Je suis plus intéressé à développer des animations sur ordinateurs pour raconter de vraies histoires plutôt que de montrer des scènes réalistes. Un grand coup de chapeau va à l'équipe Pixar pour leur excellent travail dans Toy Story. Ils ont réussi à raconter une histoire vraiment intéressante avec d'excellents graphismes. Cependant même les gens de Pixar admettent qu'ils ne peuvent pas

animer des personnes de manière convaincante. Il y a trop de subtilités dans le visage humain pour en faire des rendus adéquates sur ordinateur. Je pense que dans le futur, on verra des ordinateurs utilisés pour créer des animations qui ne seront plus aussi réalistes mais plutôt à même de raconter une histoire. Un personnage de dessin animé est très simple et graphique cependant il peut véhiculer beaucoup d'émotions avec des gestes, des poses et quelquefois des déformations extrêmes. Celles-ci ne peuvent être interprétées comme la réalité et personne ne s'attend à ce qu'elles le soient. Alors elles sont simplement acceptées pour ce qu'elles sont. Personne ne dirait que les jambes de Bugs Bunny sont trop longues ou que les lapins ne portent pas de gants blancs. Personne ne s'attend à trouver dans un lapin de dessin animé un lapin ressemblant à la réalité. Des personnages comme ceux-là ont cependant un énorme potentiel pour raconter des histoires intéressantes et amuser. Je pense que c'est une voie très intéressante pour les graphismes faits sur ordinateur. Le haut réalisme devient ennuyeux. Retournons au fun.

> Si vous désirez développer un thème ou juste donner le bonjour aux Français c'est l'occasion.

C'était très intéressant de voir des réponses de France pour DKBTrace et POV-Ray. J'ai reçu beaucoup de lettres de France (quelques unes en français) et j'ai vu des articles écrits dans des magazines français à propos de POV-Ray. Je peux usuellement lire le français très bien mais ces articles me font découvrir du nouveau vocabulaire et de nouvelles expressions. Ce genre de chose tombe bien ici au Canada. En fait, je suis couramment dans Quebec City pour donner cours et conseils en français. J'ai reçu beaucoup d'autres réponses venant de beaucoup de pays dans le monde comme l'Angleterre, l'Allemagne, l'Italie, le Japon, la Corée, la Chine, la Nouvelle-Zélande, l'Australie, la Taïwan, la Finlande et la Suède pour en

**WGirls - Mike Miller**

nommer quelques uns. J'aimerais voyager dans un de ces pays. Pour l'instant je laisse DKBTrace et POV-Ray faire le voyage à ma place.

> David K. Buck merci d'avoir accepté cette interview et bonne chance pour la suite.

NdN : Quelques passages ont été tronqués notamment à la 3ème et 9ème question...

Techno logiquement



>Correct

Les B-Splines

Quoi de plus intéressant que de pouvoir

décrire des trajectoires d'objets animés par des courbes intuitives à manipuler, ou bien obtenir des surfaces maillées avec ces mêmes courbes, comme dans le cas de visages, ou encore faire de l'interpolation pour des couleurs de textures ?

Dans un ancien numéro de la newsletter, le numéro 5 exactement, il était question des courbes de Bézier. Seulement voilà, elles ont un inconvénient majeur : lorsqu'on veut en faire des modifications cela reste un exercice de style ! En effet, en changeant un de ses points de contrôle, toute la courbe est modifiée. Et c'est là que les b-splines viennent à la rescoussent puisque qu'elles introduisent la notion de localité. Cela donne concrètement à chaque point de contrôle une influence localisée. Leur manipulation est alors plus naturelle...

A noter qu'une courbe b-spline ne passe pas au travers ses points de contrôle.

Voici comment est définie une courbe à base de b-spline pour m points de contrôle

$$Q(u) = \sum P_i B_i(u)$$

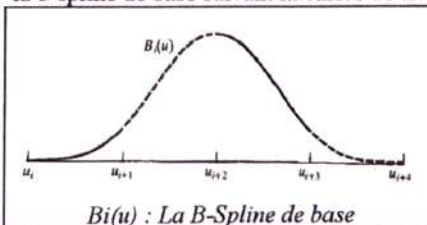
somme pour $0 \leq i < m$
avec $0 \leq u \leq l$

$Q(u)$ est la courbe complète

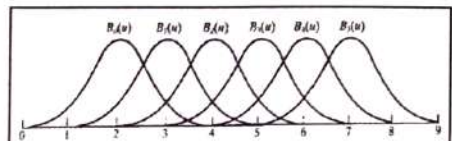
P_i sont les points de contrôle
(coordonnée x ou y)

$B_i(u)$ sont les équations associées qui permettront de donner un "poids" dans le rôle de chacun des 4 points de contrôle qui définissent alors un segment donné de la courbe \mathcal{Q} .

$Bi(u)$ est en fait un des 4 morceaux de la b-spline de base suivant la valeur de u :



Dans le cas d'une courbe Q à $m=6$ points de contrôle, on utilise 6 b-splines de base :



Splines utilisées dans le cas d'une courbe à 6 points de contrôle p_0, p_1, \dots, p_5 . A chacun de ces points, une b-spline est associée.

Pour l'intervalle $[3 ; 4]$, on a :

$$B_3 = (1/6) u^3$$

$$B_2 = (1/6) (-3u^3 + 3u^2 + 3u + 1)$$

$$B_1 = (1/6) (3u^3 - 6u^2 + 4)$$

$$B_0 = (1/6) (1-u)^3$$

Pour l'intervalle $[4; 5]$ seuls rentrent en jeu B_4, B_3, B_2, B_1 , et pour $[5; 6]$ seuls B_5, B_4, B_3, B_2 influencent la courbe Q . Leur formulation est identique aux B_i de l'intervalle $[3; 4]$.

La courbe Q est uniquement définie sur l'intervalle $i = [3 ; 6]$. Le calcul de la courbe Q pour un intervalle i donné est donné par le programme en C suivant :

```
/* Calcul de B-Spline en u */
```

```
float Compute_Spline(float p0,
float p1, float p2, float p3,
float u)
{
    float r;
    r=(1./6.)*( p0*(1-u)*(1-u)*
    (1-u) + p1*(3*u*u*u-6*u*u+4)
    + p2*(-3*u*u*u+3*u*u+3*u+1)
    + p3*u*u*u );
    return r;
}
```

Maintenant il faut pouvoir tracer une courbe entière à partir des points de contrôle. Il suffit pour cela, de parcourir la liste des points de contrôle par groupe de 4, et de tracer à l'écran au fur et à mesure.

```
/* Programme principal pour
calculer une courbe
```



```

par b-spline */

float x,y,t;
int p,n=100;

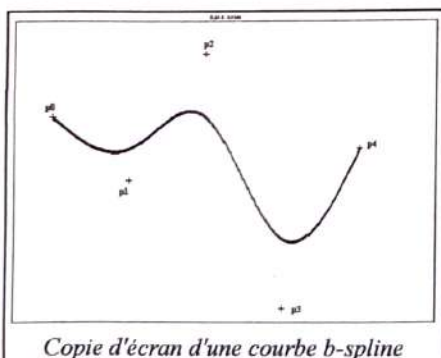
x=Compute_Spline(px[0],px[1],
px[2],px[3],0);
y=Compute_Spline(py[0],py[1],
py[2],py[3],0);
DevMove((int) x, (int) y);

/* 9 points de contrôle sont
utilisés p0, p1,..., p8 : 6
segments seront tracés */

for (p=0; p<6; p++)
{
/* Chaque segment */
for (t=0.; t<=1.0; t+=(1./n))
{
DevDraw((int) x), (int) y);
x=Compute_Spline(px[p+0],
px[p+1],px[p+2],
px[p+3],t);
y=Compute_Spline(py[p+0],
py[p+1],py[p+2],
py[p+3],t);
}
}
DevDraw((int) x,(int) y);

```

Voilà un exemple de courbe tracée avec 9 points de contrôle ; 5 sont en fait apparent, mais les premier et dernier points p0 et p4 ont été triplés de manière à ce que la courbe touche ces points (sinon elle aurait commencé plus tard et se serait terminé plus tôt) :



Un autre type de courbe à base de b-spline non-uniforme permet d'éviter d'avoir à tripler les points de contrôle pour que certains points touchent la courbe, mais la b-spline de base est alors à modifier. On peut le faire par l'algorithme récursif de Cox-DeBoor, mais c'est une autre histoire...

- Nicolas



Logo par François Gutherz

Pour recevoir WG

C'est simple et ça ne vous engage à rien : envoyez une enveloppe affranchie à 4.40F à vos noms et adresses... et un beau matin vous recevrez un papier qui se révélera être... la newsletter.

Couverture

L'image de couverture a été réalisé par **Robert A. Mickelsen** avec Persistence Of Vision.

Projet

Nom de code : D-Light

But : Créer un raytracer multi-plateformes.

Ingrédients nécessaires : Des personnes motivées sur toute machine (Amiga, PC, Atari, Macintosh, et autre).

Programmation : C standard pour le code principal et pour chaque machine une bibliothèque graphique standardisée.

Contact : Intéressé ? Ecrivez à la rédaction

Pour les cartes postales

c'est comme d'habitude...

Wild Graphics

6 avenue de la Chasse
77500 CHELLES



Articles

Wanted



Nous recherchons des collaborateurs qui pourraient participer à la newsletter de manière bénévole (puisque celle-ci est faite bénévolement). Aussi si vous pensez avoir des articles intéressants, faites en part au rédac' chef...

Sujets possibles : Image de synthèse, fractals, presse française ou étrangère, voire nouvelle technologie ou rubriques éclectiques. Ce qui vous attend : être publié dans Wild Graphics, puis sur le CD ROM de DP Tool Club (merci à eux au passage) et enfin sur le World Wide Web... A vos plumes !

Wild Graphics

Rédacteur en Chef

Nicolas Mougel

Responsable couleur

WaiYip Cheng

Pigistes (absolut hibernators)

François Gutherz

WaiYip Cheng

Stéphane Anquetil

Jacques Demare

E-Mail :

nicolas.mougel@etu.univ-compiegne.fr

Web :

http://www.univ-compiegne.fr/
~nmougel/wild.html

