



Introduction

[Introduction](#)

[Constat](#)

[Etude 1](#)

[Etude 2](#)

[Etude 3](#)

[Conclusion](#)

[Retour Index](#)

[Moonlight Atelier](#) est en version bêta, ce qui laisse entendre qu'il n'est pas encore déverminé. Deci, delà, des imperfections agaçantes continuent à errer, ce qui serait mineur si aucune conséquence fâcheuse n'en résultait. Mais tel n'est pas le cas. Il apparaît notamment qu'au chargement d'une scène précédemment sauvegardée, certains objets ont disparu. Quiconque aura été confronté à la situation sait qu'il n'en est rien, et qu'une manipulation adéquate remet les choses en ordre. Mais ça agace. Il n'y a rien à faire, ça agace même très fort!

Stefan PINKERT, sur son site [Moonlight 3D Corner](#), avait proposé une solution, qui ne semble pas donner les résultats escomptés; Gigi GIORGI, talentueux artiste et programmeur (en TCL/TK), invite à utiliser la méthode décrite ci-dessous.



Constat

[Introduction](#)

[Constat](#)

[Etude 1](#)

[Etude 2](#)

[Etude 3](#)

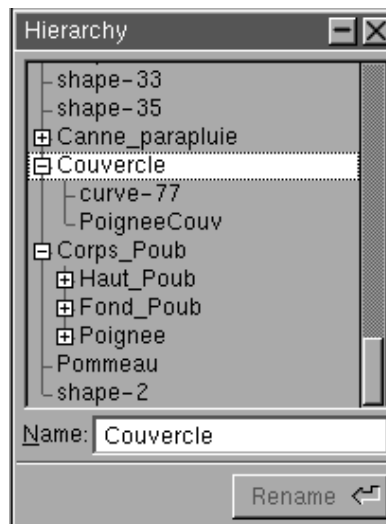
[Conclusion](#)

[Retour Index](#)

Tous les objets ne disparaissent pas; cela est vrai pour les objets **Composite Polygon**, même après conversion en **Nurbs**. A se demander d'ailleurs si la conversion a eu lieu.

Cependant, il m'est apparu également (à moi, pas à Gigi) , que certaines surfaces obtenues par **Revolution**, **Extrusion**, **Skin...** pouvaient se retrouver désorganisées lors d'un rechargement de scène. Il semblerait que hiérarchiser les objets avec **Objects> Reparent child** ou **Objects>Reparent selected children**.

Dans l'exemple suivant, l'objet Couvercle, en fait une surface, est l'objet **Parent** de la courbe Curve-77 qui a permis de le tracer, ainsi que de l'objet PoignéeCouv. L'objet Corps_Poub est **Parent** des objets fils, **Children**, Haut_Poub, Fond_Poub et Poignée. Le signe **+** devant l'objet permet de déployer son arbre de construction et de dévoiler les éléments qu'il a nécessité.



Fin de la digression. N'hésitez pas cependant à hiérarchiser vos objets et surtout à les nommer: il devient rapidement impossible de se repérer dans une liste de **shapes** et **curves** numérotés.

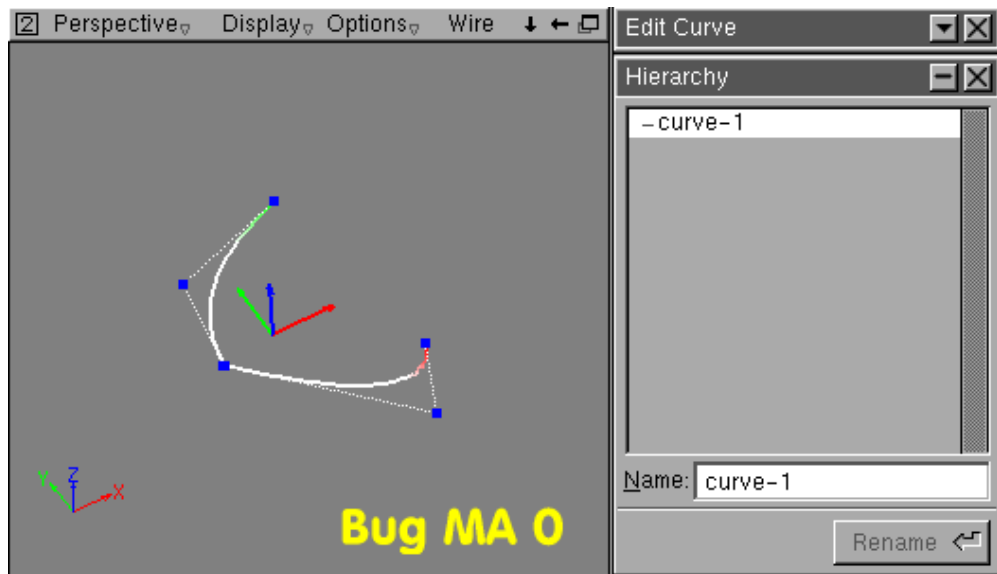


- Introduction
- Constat
- Etude 1
- Etude 2
- Etude 3
- Conclusion
- Retour Index

Etude d'un cas

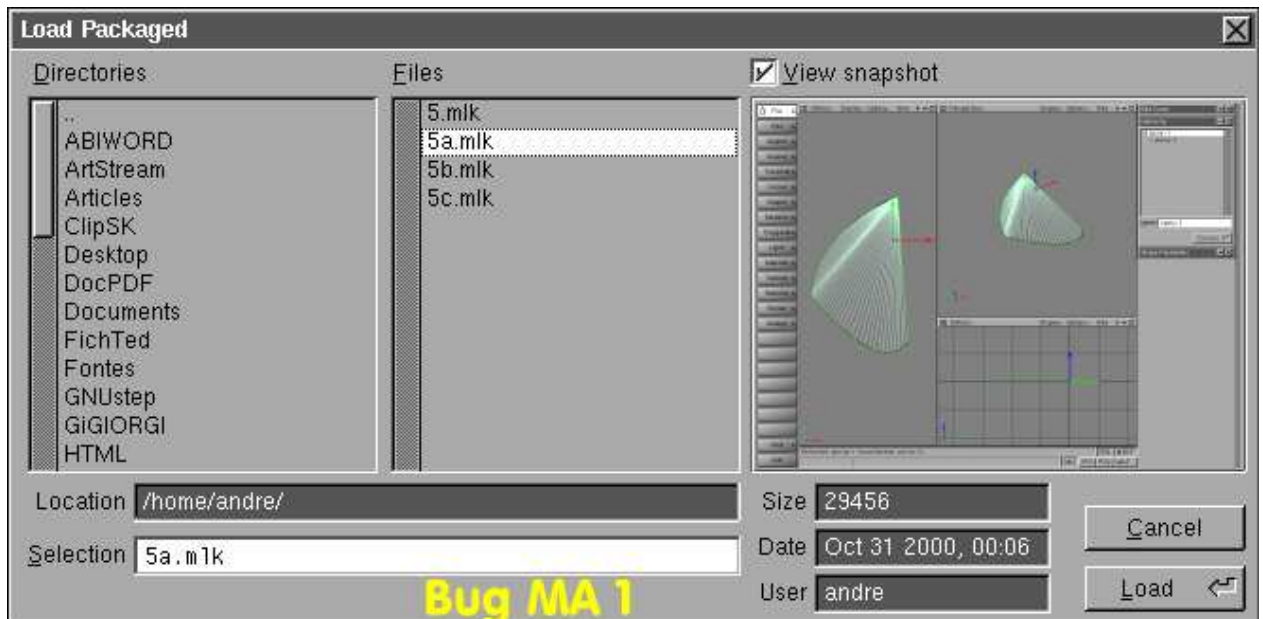
3.1 l'objet de base

Il s'agira d'une simple courbe **Nurbs**, que l'on sauvegarde sous le nom, pourquoi pas, de 5.mlk



3.2 La création du problème

A partir de la courbe précédente, créons une surface par **Relations > Composite Polygon**, que nous sauvegardons sous le nom de 5a.mlk.



L'image ci-dessus montre que 5a.mlk, grâce à la copie d'écran effectuée automatiquement (si la fonction est validée, évidemment), contient une surface.



Etude d'un cas 2

Introduction

Constat

Etude 1

Etude 2

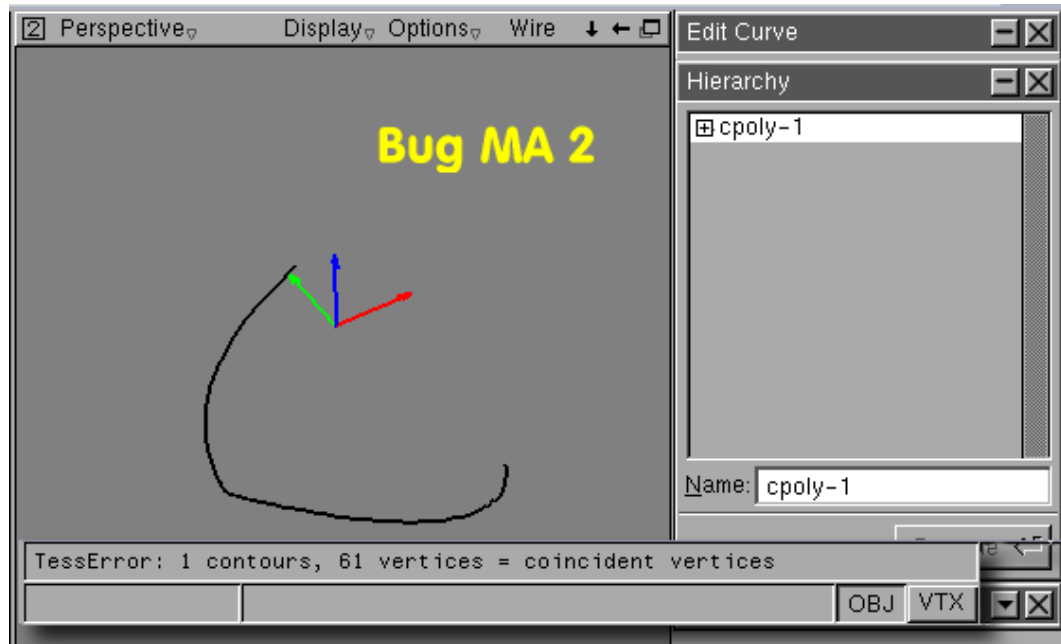
Etude 3

Conclusion

Retour Index

3.3 Chargement de la surface enregistrée

L'image précédente montrait ce que l'on était en droit d'attendre du chargement; en fait voici ce que l'on obtient: une courbe!



Simplement la courbe de base? Non: le système prévient d'une erreur rencontrée, **coïncident vertices**. Par la même occasion, il nous prévient de l'existence d'un objet constitué d'un contour, ce que l'on aurait pu voir en déployant **cpoly-1** dans **Hierarchy**, et de 61 sommets **vertices**. Donc l'objet est toujours présent dans la base de données du dessin.

En déplaçant n'importe quel point de la courbe de base par **m+clic drag**, l'objet se remet en place.



Etude d'un cas 3

[Introduction](#)

[Constat](#)

[Etude 1](#)

[Etude 2](#)

[Etude 3](#)

[Conclusion](#)

[Retour Index](#)

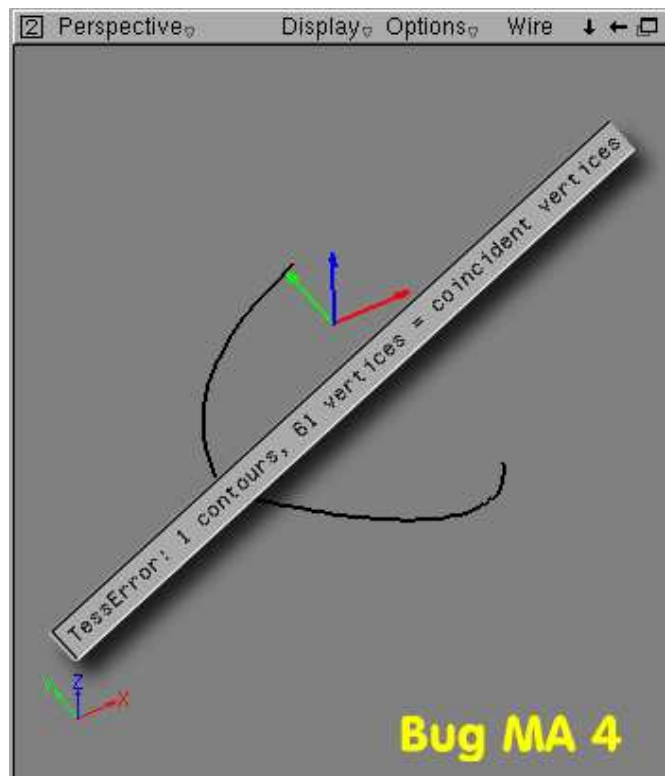
3.4 Essai de la conversion Nurbs

Si l'erreur vient de la nature même des surfaces **Composite Polygon**, convertissons l'objet en surface **Nurbs** par **Relations>Convert to Nurbs**. Le résultat est sauvegardé sous le nom de 5b.mlk, et lorsqu'on s'apprête à le recharger, **View snapshot** nous indique bien que l'objet sauvegardé est une surface.



3.5 Vérification

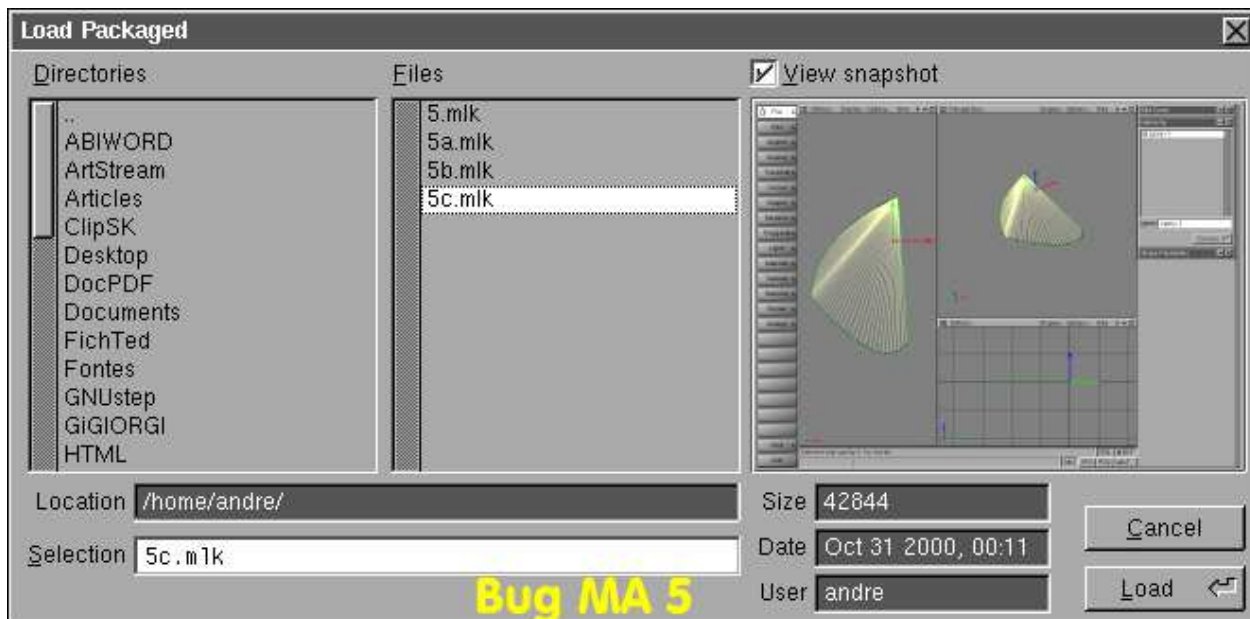
Il suffit donc de recharger 5b.mlk pour s'apercevoir ... que le vilain méchant bogue a encore frappé!



**Etude d'un cas 4 et conclusion**[Introduction](#)[Constat](#)[Etude 1](#)[Etude 2](#)[Etude 3](#)[Conclusion](#)[Retour Index](#)**3.6 Mais alors que faire?**

Quelque chose d'insensé: l'objet **Composite Polygon** est converti en **Nurbs**, puis la surface **Nurbs** est de nouveau convertie en **Polygon**.

Pour ce faire, bien s'assurer que l'on se trouve en mode simple objet **SGL**, et sauvegarder sous le nom de 5c.mlk



Vérifier maintenant que l'on ne se trouve pas en état de péché mortel, croiser les doigts, caresser son grigri préféré, et recharger la scène 5c.mlk. **Miracle!** cela fonctionne! Alléluia!

4 Conclusion

Elle sera courte mais pertinente: l'informatique, ce n'est qu'une affaire de logique rigoureuse ainsi que cela vient d'être démontré brillamment. Heu: n'en voulez pas à Gigi pour le ton badin, voire irrévérencieux. Il n'y est pour rien et je j'ai juste voulu m'amuser.

