



- **Introduction**
- **Partie 1**
- **Partie 2**
- **Partie 3**
- **Partie 4**
- **Partie 5**
- **Partie 6**
- **Partie 7**
- **Partie 8**
- **Partie 9**
- **Partie 10**
- **Partie 11**
- **Conclusion**
- **Retour Index**

Introduction

Allez savoir pourquoi, dès qu'un programme 3D est présenté, il y a toujours une modélisation de théière comme preuve de ses capacités à reproduire convenablement la réalité. Même le grand 3D Studio n'échappe pas à la règle. De fait, lorsque j'ai commencé la 3D ludique avec Rhinocéros sous Windows, une des premières scènes que j'ai composée contenait une théière, ajoutant ainsi un exemplaire supplémentaire aux millions d'ustensiles de ce type déjà modélisés.



De fait, quel que soit le contenu de la scène, théière ou non, un certain nombre de fonctions seront toujours utilisées:

- les basiques **Plan**, **Extrusion**, **Revolution**
- les sophistiquées **Tube/Pipe**, **Skin**, **Loft**, **Sweep**, **Bi-rail**
- les éclairages **Point Light**, **Spot Light**, **Directional Light** avec les nuances **Global** et **Indirect**, lorsqu'il y a possibilité d'illumination, **Radiosity**
- les ombrages simples **Shadows** ou doux **Soft Shadows**
- les texturages de surface **Mapping** avec bosselage ou non **Bump**
- les caractéristiques de matériaux **Transparency**, **Diffuse**, **Emission**, **Speculaire**, **Reflection**, **Refraction**
- les fonctions **booléennes**, absentes de Moonlight qui est surfacique et non volumique
- les constructions par **primitives** et courbes de **Bézier** et **Nurbs**.

Aussi, peu importe la scène que l'on construit si l'objectif est l'apprentissage, puisque certaines méthodes seront immuables.

Nous allons donc apprendre à modéliser une théière avec **Moonlight**. Toutes les fonctions qui viennent d'être énoncées ne seront pas utilisées et l'on supposera que l'interface du programme est connue, de même que le paramétrage des matériaux et du module de rendu.

Si ce n'est pas le cas, prière de se reporter aux didacticiels spécifiques disponibles sur [Linuxgraphic](http://Linuxgraphic.org).

Au chapitre Conclusion, on peut déjà voir le résultat attendu du labeur décrit plus bas.



-
- [Introduction](#)
- [Partie 1](#)
- [Partie 2](#)
- [Partie 3](#)
- [Partie 4](#)
- [Partie 5](#)
- [Partie 6](#)
- [Partie 7](#)
- [Partie 8](#)
- [Partie 9](#)
- [Partie 10](#)
- [Partie 11](#)
- [Conclusion](#)
- [Retour Index](#)

Partie 1

Le bec verseur

Chacun commencera par l'élément de l'objet qui lui convient le mieux. Mais la difficulté dans la modélisation d'une théière (ou d'une cafetière, d'un broc...) réside dans le bec verseur. On utilisera pour ce faire la fonction **Skin surface**.

Le principe en est simple. Imaginez une coque de bateau ou une carlingue d'avion: toutes deux sont constituées d'une "peau" tendue sur des formes réparties sur la longueur de l'objet. Ces formes sont appelées selon leur champ d'utilisation: couples membrures, section... Ici, je dirai: section.

La fonction **Skin surface** est très souple: il suffit de disposer les sections de dimensions et de forme voulues tout au long de l'axe imaginaire du bec verseur, et d'y tendre une peau. Le plus simple est de tracer d'abord ces sections sous forme de cercles que l'on modifiera ensuite à son gré.

Petit rappel:

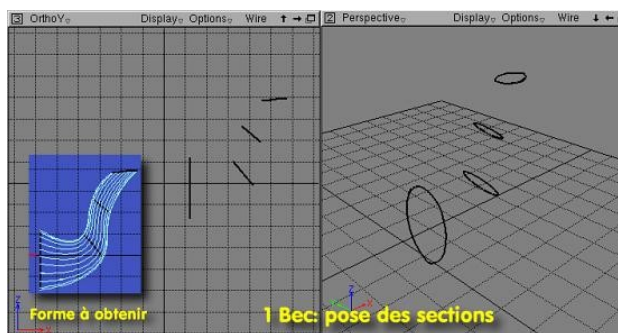
Pour modifier l'échelle d'un objet (courbe ou surface): le sélectionner par **espace+clic**, puis en maintenant la touche **x** enfoncée, déplacer la souris

Pour modifier la forme d'un objet: le sélectionner, appuyer sur **m**, déplacer à la souris le point de contrôle voulu.

Pour déplacer l'objet: le sélectionner, appuyer sur **v** et déplacer la souris

– Disposer les sections supports

Il est bien évident qu'il faut avoir à l'esprit la forme de l'objet que l'on veut générer; on pourra s'appuyer sur une courbe Nurbs qui figurera l'axe, afin de travailler avec précision.





Articles

- [Introduction](#)
- [Partie 1](#)
- [Partie 2](#)
- [Partie 3](#)
- [Partie 4](#)
- [Partie 5](#)
- [Partie 6](#)
- [Partie 7](#)
- [Partie 8](#)
- [Partie 9](#)
- [Partie 10](#)
- [Partie 11](#)
- [Conclusion](#)
- [Retour Index](#)

Partie 2

Le bec verseur (suite)

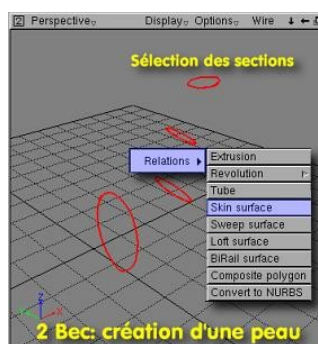
– *Tendre une peau*

Ceci se fait en trois étapes:

– choisir la fonction >**Relations**>**Skin surface**

– sélectionner les sections (dans l'ordre: de la première vers la dernière suivant l'axe de l'objet, ou de la dernière vers la première)

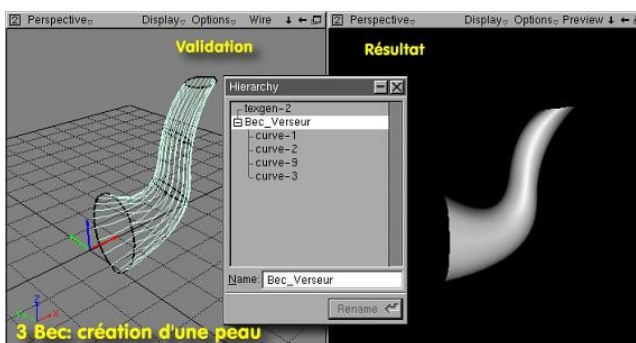
La sélection s'effectue simplement en cliquant sur les courbes, qui deviennent rouges



– Valider le choix

Ne pas oublier de regarder les informations contenues dans la zone de statut, en bas de l'écran; il y est dit L(pick/unpick) M(unpick) R(Validate), ce qui correspond aux actions obtenues par les boutons gauche, milieu et droite de la souris.

Donc quand la sélection est complète, valider par **clik droit**: la surface est créée instantanément



Dans la boîte de dialogue **Hierarchy** déposée sur l'établi, on changera le nom de Skin-1 donné par défaut à la forme créée par un nom plus explicite, ce qui permet, lorsque la scène comporte beaucoup d'objets et qu'il devient difficile d'en sélectionner un précisément, d'effectuer la sélection en cliquant sur le nom de l'objet dans cette boîte. On remarquera par ailleurs que lorsqu'on déplie l'entrée Skin-1, ici renommée Bec_Verseur, le détail des sections sur lesquelles la peau est tendue.

Remarque importante: si la forme ne convient pas, on peut sélectionner une section et la déplacer, ou la modifier en forme ou en dimension. L'objet est réactualisé en temps réel, tant en filaire (**Wire**) qu'en

ombrage (Preview)



Articles

- Introduction
- [Partie 1](#)
- [Partie 2](#)
- [Partie 3](#)
- [Partie 4](#)
- [Partie 5](#)
- [Partie 6](#)
- [Partie 7](#)
- [Partie 8](#)
- [Partie 9](#)
- [Partie 10](#)
- [Partie 11](#)
- Conclusion
- Retour Index

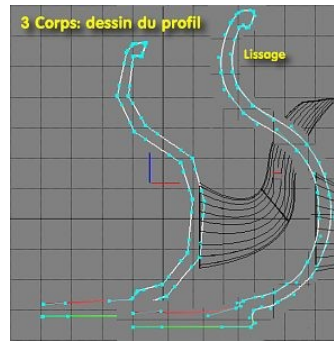
Partie 3

Le corps de la théière

On le construira par **Revolution** d'un profil autour d'un axe.

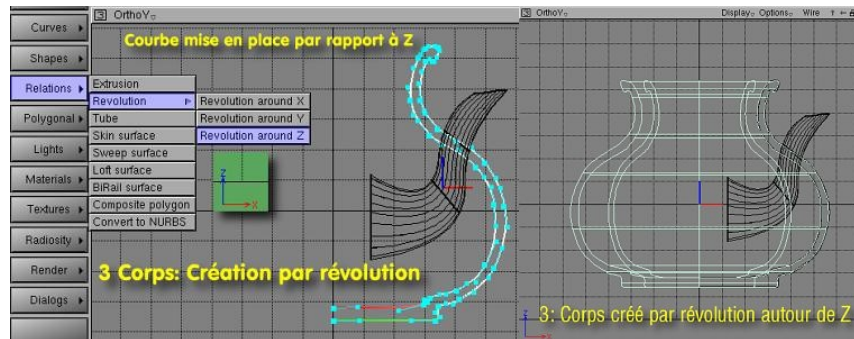
– Dessin du profil

Celui-ci peut être dessiné à l'aide de **>Curves> Bezier curve**, ou **NURBS curve**, ou **Bezier polyline**. Pour ma part, je préfère utiliser une polyligne, plus facilement contrôlable, que l'on lisse en manipulant ses points et ses tangentes. Par la suite, cette courbe sera convertie en Nurbs par **>Curves> Convert Bezier/NURBS**

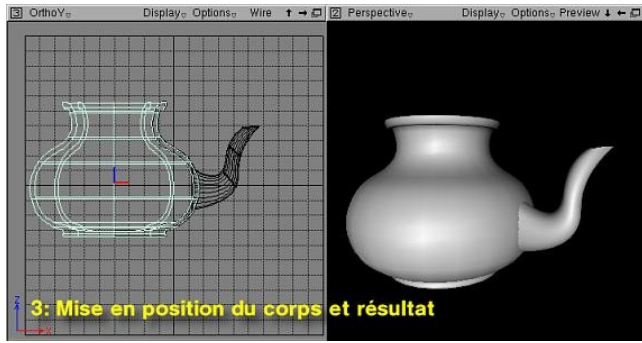


– Génération du corps

Bien repérer la direction de l'axe de révolution de la courbe; ici, il est parallèle à l'axe Z. De plus, la révolution s'effectuera autour de l'axe du repère global qui passe par le centre de la fenêtre: il est donc impératif de déplacer le profil est de mettre son axe virtuel en coïncidence avec l'axe Z absolu (à moins qu'il n'y ait une méthode pour définir un axe de révolution local; en ce cas je suis preneur de la solution)



Quand le volume est généré, il suffit de le positionner par rapport au bec verseur. Cette démarche sera à utiliser pour tous les objets obtenus par révolution.





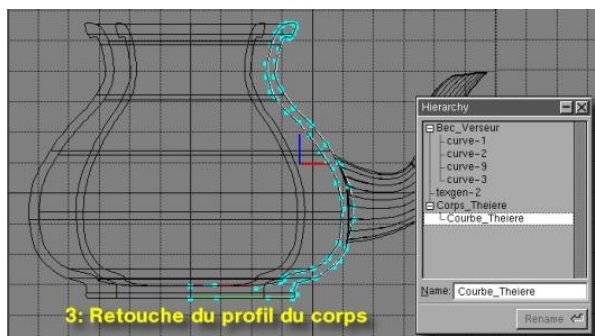
Articles

- [Introduction](#)
- [Partie 1](#)
- [Partie 2](#)
- [Partie 3](#)
- [Partie 4](#)
- [Partie 5](#)
- [Partie 6](#)
- [Partie 7](#)
- [Partie 8](#)
- [Partie 9](#)
- [Partie 10](#)
- [Partie 11](#)
- [Conclusion](#)
- [Retour Index](#)

Partie 4

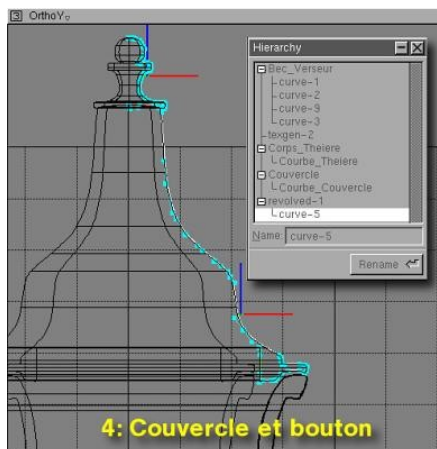
Modification du profil (si nécessaire)

On notera qu'en éditant encore une fois le profil, on modifiera le volume en temps réel: c'est en cela que **Moonlight** mérite le nom de modelleur. On a vraiment l'impression de travailler de la pâte à modeler
Remarque importante: il y a une restriction à ce qui vient d'être dit. Si l'on modifie un profil alors qu'un calcul d'illumination **Radiosity** est en cours, c'est le crash assuré. Prudence!



Création du couvercle et du bouton

Par la même méthode que celle décrite précédemment, construire le couvercle et le bouton chapeautant le tout.





Articles

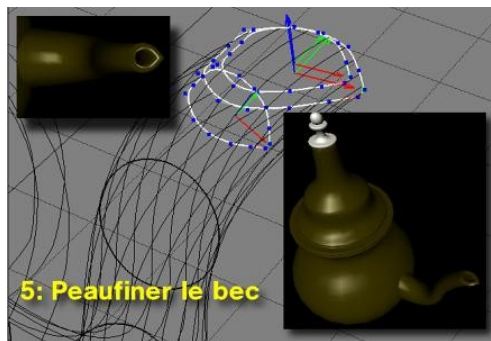
- [Introduction](#)
- [Partie 1](#)
- [Partie 2](#)
- [Partie 3](#)
- [Partie 4](#)
- [Partie 5](#)
- [Partie 6](#)
- [Partie 7](#)
- [Partie 8](#)
- [Partie 9](#)
- [Partie 10](#)
- [Partie 11](#)
- [Conclusion](#)
- [Retour Index](#)

Partie 5

Peaufiner le bec verseur

Moonlight étant un programme surfacique, les objets n'ont pas d'épaisseur, notamment lorsqu'ils sont élaborés par **Skin**. C'est le cas du bec. Si donc on le visualise en plongée, un effet disgracieux apparaîtra. Il convient alors de lui donner de l'épaisseur, afin de le rendre plus réaliste.

Pour cela, il suffit de dupliquer la section extrême, de la réduire par changement d'échelle, et de la positionner au bon endroit. Ensuite, suivront une duplication de cette nouvelle section et son positionnement à l'intérieur du bec. Une peau sera tendue entre ces trois sections pour obtenir le résultat suivant:





- Introduction
- [Partie 1](#)
- [Partie 2](#)
- [Partie 3](#)
- [Partie 4](#)
- [Partie 5](#)
- [Partie 6](#)
- [Partie 7](#)
- [Partie 8](#)
- [Partie 9](#)
- [Partie 10](#)
- [Partie 11](#)
- Conclusion
- [Retour Index](#)

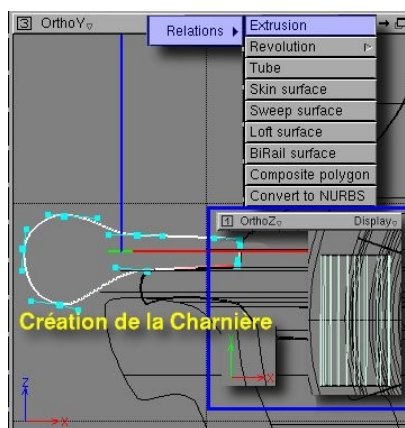
Partie 6

Réaliser une charnière

On supposera que le couvercle pivote sur une bordure de la théière grâce à une charnière située à l'opposée du bec verseur. Pour ce faire on utilisera deux fonctions: **Extrusion** et **Composite polygon**, cette dernière étant très puissante pour obtenir des surfaces gauches.

Le principe de l'extrusion est simple: il s'agit de créer un profilé à partir d'une forme plane, dont la direction est obligatoirement perpendiculaire au plan sur lequel est tracé la forme, c'est-à-dire le profil. Si donc on dessine un profil dans le plan XY, l'extrusion (le profilé) se fera suivant l'axe Z.

– Le profilé



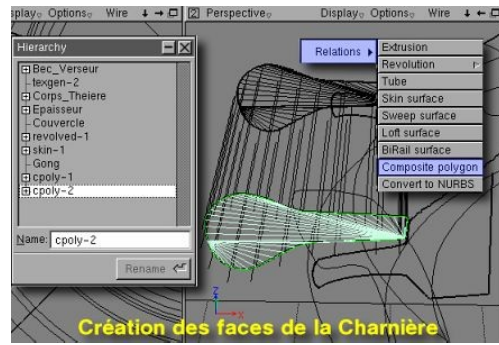
Dans l'exemple qui nous occupe, le profil est dessiné dans le plan XZ avec une polygline de Bézier que l'on lissera à son gré en agissant sur les points de contrôle et les tangentes. Quand le tracé paraît convenable, la commande **>Relation>Extrusion**, génère le profilé suivant l'axe Y, perpendiculaire au plan de tracé. La longueur du profilé est donnée par défaut: il suffit, pour la régler à ses besoins, de sélectionner une extrémité du profilé et de la déplacer à la souris (ou par l'intermédiaire de la boîte de dialogue **Transformations**) dans le sens et de la valeur voulus. On devrait obtenir quelque chose ressemblant à l'image ci-dessus.

– Les faces

Moonlight étant, encore une fois, surfacique, ce que l'on vient d'élaborer n'a ni épaisseur ni faces en bout: il s'agit d'un tube de section non circulaire. Il convient donc d'apposer deux faces en bout pour clore l'objet.

L'opération consiste à sélectionner **>Relations> Composite polygon** et à désigner la courbe qui limitera la surface à produire. Celle-ci sera plane si la courbe est tracée dans un plan; sinon, la surface sera gauche.

Comme pour les autres objets, il est toujours possible d'éditer la courbe génératrice et d'en modifier l'aspect, ce qui entraînera une mise à jour automatique de la surface.



On notera que par défaut, la surface ainsi créée s'appelle cpoly-x dans **Hierarchy**, et que son entrée se déplie pour laisser apparaître curve-x comme composant. Il sera judicieux là encore de renommer plus explicitement cpoly-x.



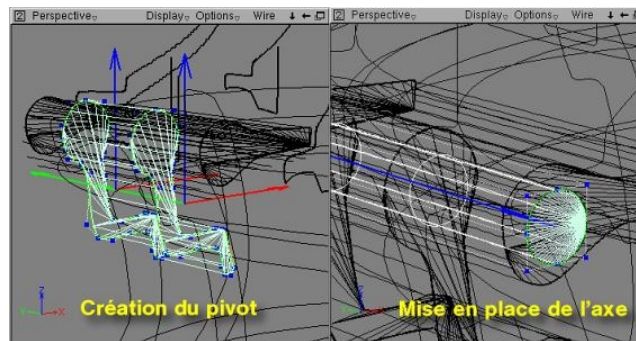
- [Introduction](#)
- [Partie 1](#)
- [Partie 2](#)
- [Partie 3](#)
- [Partie 4](#)
- [Partie 5](#)
- [Partie 6](#)
- [Partie 7](#)
- [Partie 8](#)
- [Partie 9](#)
- [Partie 10](#)
- [Partie 11](#)
- [Conclusion](#)
- [Retour Index](#)

Partie 7

Le Pivot et l'axe

Une charnière s'articule sur une partie fixe, qu'on appellera pivot même si le nom n'est pas adéquat, et autour d' un axe qui traversera l'ensemble.

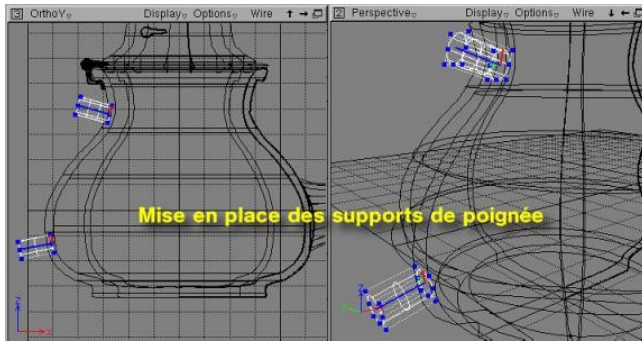
Le pivot sera construit exactement comme la charnière précédente. Donc pas de commentaire spécifiques pour le pivot.



L'axe sera bâti à partir **>Shapes>NURBS>Cylinder** ou, si l'on préfère, **>Shapes>Parametric>Cylinder**. Dans les deux cas, le cylindre est proposé avec un rayon et une longueur prédéfinis qu'il faudra adapter, soit par changement d'échelle, ce que l'on fera par le biais de la boîte de dialogue **Modes List** qui permet de sélectionner un changement d'échelle directionnel (suivant un seul axe: **Scale XYZ**), soit par le biais de la boîte de dialogue **Shape Primitives**, qui permet de paramétrer rayon de départ, rayon d'arrivée, longueur, ce qui autorise d'ailleurs la transformation d'un cylindre en tronc de cône ou en cône. Bien entendu, il faudra donner une surface d'extrémité au cylindre grâce à **Composite polygon**. Personnellement, j'ai utilisé **NURBS Cylinder**, et adapté les dimensions à la souris. Pour entrer des valeurs numériques, il faut savoir que l'unité de mesure est fonction du paramétrage de la grille, la valeur 1 représentant un carreau de celle-ci.

Les supports de poignée

La poignée n'est pas directement accolée au corps de la théière, en tout cas sur l'objet réel qui me servait de modèle. Elle repose sur deux excroissances cylindriques moulées avec le corps. Deux cylindres définis comme l'axe précédent conviendront parfaitement. Mais soyons rusés: dès qu'un cylindre convenable est mis en place, on l'utilisera pour définir l'autre par duplication, afin qu'ils soient identiques. Utiliser pour cela **>Objects>Duplicate** ou simplement la touche **d+clac** sur l'objet, qui se trouve reproduit exactement sur l'objet source. La touche **v+** déplacement de souris le positionnera ailleurs.





- [Introduction](#)
- [Partie 1](#)
- [Partie 2](#)
- [Partie 3](#)
- [Partie 4](#)
- [Partie 5](#)
- [Partie 6](#)
- [Partie 7](#)
- [Partie 8](#)
- [Partie 9](#)
- [Partie 10](#)
- [Partie 11](#)
- [Conclusion](#)
- [Retour Index](#)

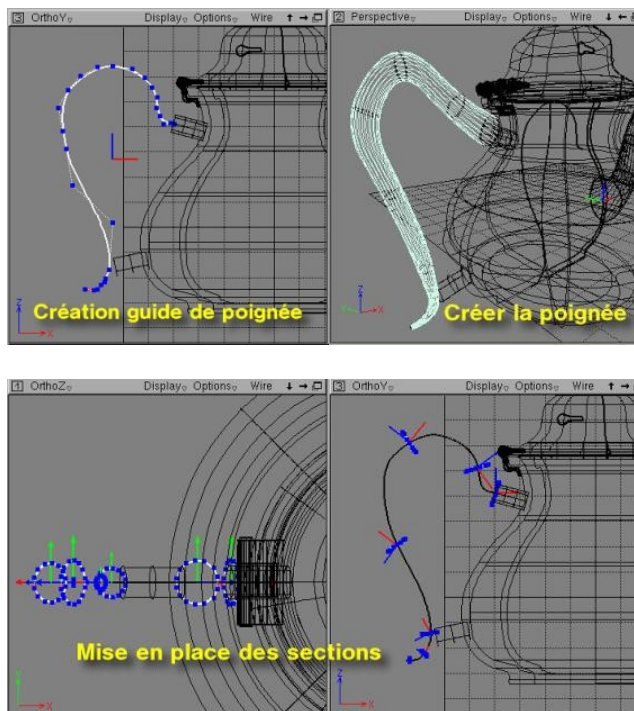
Partie 8

La poignée

On aurait pu utiliser une fonction simple en vue de l'obtenir: **>Relations>Tube**. L'inconvénient de cette commande est de produire un objet certes sinueux (ou non, selon le guide dessiné) mais de section constante: un vrai tube donc. Cependant, il est plus esthétique de réaliser une poignée de section variable, comme sur la poignée réelle d'ailleurs. Une nouvelle fois, nous allons recourir à la fonction **Skin surface**, selon la méthode utilisée pour obtenir le bec verseur.

Trois étapes sont nécessaires:

- dessin de l'axe guide
- pose des sections
- pose de la peau



L'illustration montre les différentes tailles des sections, mais il est bon de savoir qu'elles peuvent également avoir des forme différentes. Rien n'est plus facile que de commencer avec une section en étoile pour terminer avec une section en ellipse, par exemple. La peau sera convenablement tendue.



Articles

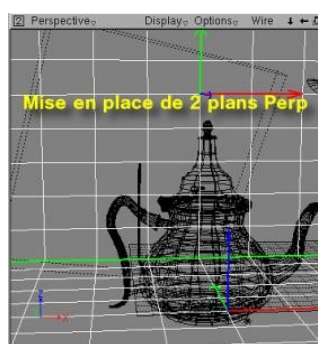
- [Introduction](#)
- [Partie 1](#)
- [Partie 2](#)
- [Partie 3](#)
- [Partie 4](#)
- [Partie 5](#)
- [Partie 6](#)
- [Partie 7](#)
- [Partie 8](#)
- [Partie 9](#)
- [Partie 10](#)
- [Partie 11](#)
- [Conclusion](#)
- [Retour Index](#)

Partie 9

L'environnement

– Sol et mur

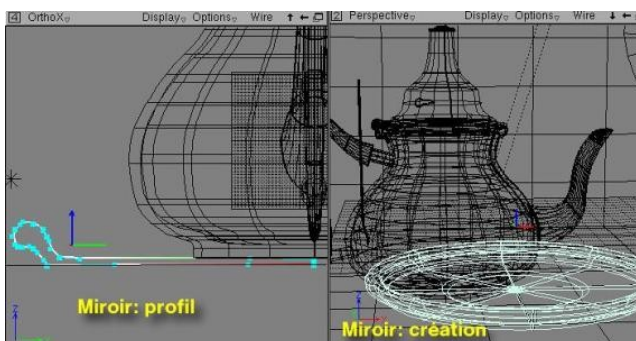
Cette théière ne lévite pas dans un univers noir; elle doit reposer sur une surface, plane ou non, limitée par n'importe quoi, puisque l'infini ne peut être représenté. Deux plans perpendiculaires répondrons parfaitement à ce besoin. Je préconise >Shapes>Paramétric>Plane plutôt qu'un plan NURBS. En effet, certains problème de distorsion et d'incohérence de comportement me sont apparus lorsqu'un plan Nurbs était exagérément agrandi par changement d'échelle.



Le centre du plan déposé coïncide avec l'origine du repère absolu (global), donc avec le centre de la fenêtre. Si les deux plans doivent avoir les mêmes caractéristiques de taille et d'attributs, il faut dimensionner ce premier plan et définir son matériau et sa texture. Il suffira alors de le dupliquer par >Objets>Instantiate pour obtenir une copie conforme qu'on mettra en place par une rotation à 90° et une translation, déplacements contrôlables dans la boîte de dialogue **Transformation** déposée sur l'établi.

– Le miroir et les éclairages

Il est intéressant de voir un objet sous toutes ses coutures en une seule vue, surtout s'il n'est pas symétrique, ce qui n'est d'ailleurs pas le cas de la théière. Pour cela rien de tel que de placer un miroir qui en montrera la face cachée. Arbitrairement, on choisira un miroir circulaire réalisé par **Revolution**.





Articles

- Introduction
- [Partie 1](#)
- [Partie 2](#)
- [Partie 3](#)
- [Partie 4](#)
- [Partie 5](#)
- [Partie 6](#)
- [Partie 7](#)
- [Partie 8](#)
- [Partie 9](#)
- [Partie 10](#)
- [Partie 11](#)
- [Conclusion](#)
- [Retour Index](#)

Partie 10

L'environnement (suite)

– *Le miroir et les éclairages* (suite)

Cette théière ne lévite pas dans un univers noir; elle doit reposer sur une surface, plane ou non, limitée par n'importe quoi, puisque l'infini ne peut être représenté. Deux plans perpendiculaires répondrons parfaitement à ce besoin. Je préconise >Shapes>Paramétric>Plane plutôt qu'un plan NURBS. En effet, certains problème de distorsion et d'incohérence de comportement me sont apparus lorsqu'un plan Nurbs était exagérément agrandi par changement d'échelle.

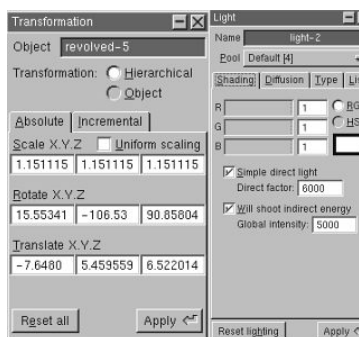
Le miroir sera disposé à l'arrière de la théière et légèrement incliné afin qu'il produise une réflexion plongeante.

La scène devra être éclairée: trois Spot light et un Point Light ont été ici utilisés. A chacun de définir l'énergie issue des lampes, et de déterminer s'il convient de leur donner l'attribut **Will shoot indirect energy**, utilisé par le calcul de radiosité



Lorsque les éclairages sont représentés en vert, c'est que l'attribut **indirect energy** est activé, ce qui est le cas pour les spots 2 et 3 et le point 4. Le spot 1 est déclaré en **Simple direct light**.

Les réglages d'éclairage, comme les déplacements rigoureux des objets s'effectuent dans des boîtes de dialogues spécifiques déposées sur l'établi. On remarquera que les transformations acceptent des valeurs à 6 décimales, que celles-ci dépendent du repère global, **Hierarchical**, ou local, **Object**. On remarquera encore qu'elles sont référencées par rapport à l'origine absolue, **Absolute**, ou relative au dernier point, **Incremental**... On remarquera aussi quatre onglets pour paramétrer un éclairage: beaucoup d'expérimentations sont à faire!





Articles

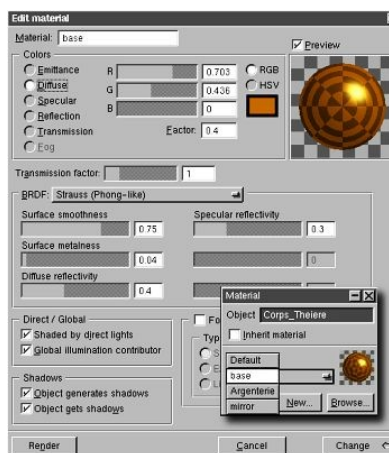
- Introduction
- [Partie 1](#)
- [Partie 2](#)
- [Partie 3](#)
- [Partie 4](#)
- [Partie 5](#)
- [Partie 6](#)
- [Partie 7](#)
- [Partie 8](#)
- [Partie 9](#)
- [Partie 10](#)
- [Partie 11](#)
- [Conclusion](#)
- [Retour Index](#)

Partie 11

Les matériaux

Par matériaux, on entend les caractéristiques de couleurs, réflexion, transparence, reflets... qui donnent son aspect à une surface.

Lors de la création d'un objet, celui-ci est affecté d'un matériau par défaut, de couleur grise, appelé **Inherit material**. Dès que la case Inherit est décochée, on peut éditer (**Edit**), créer (**New**) ou charger (**Browse**) un matériau. Dans les deux premiers cas, une boîte de dialogue s'ouvre telle que ci-dessous:



Le paramétrage du matériau est alors permis; je ne reviendrai pas sur la façon de procéder, décrite dans un didacticiel dédié, en ligne sur Linuxgraphic. Simplement, je donnerai les attributs des matériaux utilisés pour obtenir l'image visible en conclusion

Le matériau Base, qui affectent toute les objets de la théière, sauf la charnière et l'axe, est tel que:
BRDF (méthode de rendu): Strauss qui semble particulièrement convenir pour les surfaces métalliques;
voir images ci-dessus

Emittance 0

Diffuse: R0.703 G0.436 B0 Factor 0.4

Specular:R1 G1 B0.653 Factor 1

Reflection: R0.703 G0.436 B0 Factor 1

Transmission:0

Le materiau Argenterie de la charnière:

BRDF: Strauss

Emittance:0

Diffuse: 02 02 02 Factor1

Specular: H0 S0 V1 Factor 1

Reflection: H0 S0 V1 Factor 0.8

Le materiau Mirror du miroir

BRDF: default

Emittance:0

Diffuse:0
Specular:0
Reflection: H0 S0 V0.931 Factor 1

Le materiau des plans:

BRDF: default

Emittance:0

Diffuse: H0 S0 V0.8 Factor 1

Specular: H0 S0 V0.2 Factor 1

Reflection: H0 S0 V0.45 Factor 1

Texture: create UV, blend Diffuse, blend Specular, blend reflection, bump 0

La texture est un bitmap représentant du marbre.

Et c'est tout! Il ne reste plus qu'à lancer le rendu en demandant **Soft shadows** et **Antialiased**, à la taille de sortie désirée



-
- [Introduction](#)
- [Partie 1](#)
- [Partie 2](#)
- [Partie 3](#)
- [Partie 4](#)
- [Partie 5](#)
- [Partie 6](#)
- [Partie 7](#)
- [Partie 8](#)
- [Partie 9](#)
- [Partie 10](#)
- [Partie 11](#)
- [Conclusion](#)
- [Retour Index](#)

Conclusion

L'image résultante devrait parler d'elle-même, bien que fortement dégradée par la compression JPEG.



Qu'ajouter, sinon: vous prendrez bien un petit thé, avant de commencer ?

André PASCUAL
andre@linuxgraphic.org