

Java3D et Swing

par [Romain Guy](#)

Date de publication :

Dernière mise à jour :

Tutoriel d'introduction sur l'utilisation de l'API Java3D avec la librairie graphique Swing.

I - Démonstration avec Java Web Start

II - Explications

III - Téléchargements

I - Démonstration avec Java Web Start



Prérequis : Pour essayer cette démonstration vous devez disposer de [Java Web Start](#) (compris avec le JRE et le JDK).

Vous devez aussi installer [Java3D](#), téléchargeable [ici](#).



[Lancez la démonstration](#)

II - Explications

Vous êtes vous jamais demandé comment utiliser de jolis composants 3D avec Swing ? Java3D est la solution. Soit, à première vue, Java3D ne vous permet pas d'intégrer facilement des scènes 3D dans vos UI. Si vous avez vu les vidéos de démonstration du produit Avalon de Microsoft, vous voyez de quoi je parle. Avalon ne sera pas disponible avant plusieurs mois, mais Java et Java3D, eux, sont déjà là.

Grâce à une simple astuce, vous pouvez créer des applications Swing stupéfiantes. Voici un exemple :



AmazonPick

Notez le dégradé du background, que j'ai mis afin de montrer l'union entre Swing et Java3D. Avant de rentrer dans les détails, regardons ce dont est capable cette application. Je l'ai appelée *AmazonPick* car elle serait une superbe interface pour une application de recherche de livres. Les livres correspondant à votre recherche seraient affichés sous forme de boutons en bas de la fenêtre (dans cet exemple, les boutons sont codés en dur). Quand vous cliquez sur l'un d'entre eux, le livre en 3D tourne pour montrer la nouvelle couverture sur l'autre face. Du plus bel effet.



Pour réaliser cela, j'ai simplement créé une texture Java3D sur laquelle j'ai dessiné le background de la fenêtre. Voici le code source, *xpanel* étant le content pane de la fenêtre et *c3d* le Canvas3D sur lequel la scène est dessinée :

```
BufferedImage image = new BufferedImage(xpanel.getWidth(),
                                         xpanel.getHeight(),
                                         BufferedImage.TYPE_INT_RGB);
getContentPane().paint(image.getGraphics());

BufferedImage subImage = new BufferedImage(CANVAS3D_WIDTH,
                                           CANVAS3D_HEIGHT,
                                           BufferedImage.TYPE_INT_RGB);
((Graphics2D) subImage.getGraphics()).drawImage(image, null,
                                                  -c3d.getX(), -c3d.getY());

Background bg = new Background(new ImageComponent2D(ImageComponent2D.FORMAT_RGB, subImage));

BoundingSphere bounds = new BoundingSphere();
bounds.setRadius(100.0);
bg.setApplicationBounds(bounds);

BranchGroup objRoot = new BranchGroup();
objRoot.addChild(bg);
```

Bien que cette astuce soit facile à implémenter, elle requiert qu'une texture du background soit réalisée. Dans notre cas, cela veut dire que l'on doit d'abord montrer la fenêtre. L'utilisateur verra donc un rectangle gris pendant un instant.

L'utilisation de Java3D a beaucoup d'avantages mais requiert quelques astuces pour fonctionner parfaitement. Un Canvas Java3D est un composant lourd (*heavyweight*) alors que Swing contient des composants légers (*lightweight*). Le résultat est que les composants Swing seront dessinés sous le canvas Java3D. Les équipes Swing et Java3D sont au courant de ce problème et il existe une [solution simple](#).

Avec Java3D 1.3.1 une meilleure solution existe. Vous pouvez utiliser un Canvas3D pour réaliser le rendu offscreen. C'est à dire que vous pouvez faire tout le travail de rendu dans un buffer et dessiner directement le résultat dans un composant Swing (avec un JComponent personnalisé ou une ImageIcon). Naturellement, cela nécessite un peu plus de travail.

Vous pouvez télécharger cette démonstration et ses sources afin de voir comment réaliser cela. Vous aurez besoin de Java3D 1.3 et J2SE 1.4 (peut-être moins, une version supérieure est bien aussi). Si vous avez seulement J2SE 1.4, vous pouvez changer une ou deux lignes de code dans *BooksDemo.java* pour vous débarrasser des generics et recompiler le tout tranquillement.

[Faites moi savoir](#) si vous avez utilisé cette astuce avec succès.

III - Téléchargements

[Version PDF](#) ([mirroir HTTP](#))

[Version HTML/ZIP](#) ([mirroir HTTP](#))

[Code source](#) ([mirroir HTTP](#))