Cahiers GUTenberg

JPDRAW, UN ÉDITEUR DE DESSINS POUR LaTeX SUR SUN

Jean-Pierre Merlet

Cahiers GUTenberg, nº 2 (1989), p. 54-62.

<http://cahiers.gutenberg.eu.org/fitem?id=CG_1989___2_54_0>

Cahiers GUTenberg
n° 2 Mai 1989

JPdraw , un éditeur de dessins pour LaTeX sur SUN

Jean-Pierre MERLET

INRIA
Centre de Sophia-Antipolis
2004 Route des Lucioles
06565 Valbonne
E-mail:merlet@alioth.inria.fr

Abstract

Nous décrivons l'utilisation d'un éditeur de dessins fonctionnant sur SUN dans l'environnement suntools. Le but premier de cet éditeur est de générer des fichiers pour l'éditeur de texte LaTeX sous la forme d'un environnement «picture ». Mais il est aussi possible de sauver les dessins en Postscript ou sous la forme d'images, ou enfin sous forme d'instructions pour un {em plotter de type Hewlett-Packard pour la création de transparents couleurs.

JPdraw permet la manipulation à la souris de primitives géométriques variées (lignes, textes, rectangles, polygones, ellipses, ovales, courbes quelconques) de manière similaire au Macdraw de Mac-Intosh. Mais il dispose aussi d'un méta-langage pour la description de dessins permettant ainsi une interface commode entre un programme utilisateur (par exemple CAO, traitement d'image) et LaTeX.

1. Introduction

L'insertion de dessins dans un source LaTeX est un problème bien connu des utilisateurs de ce logiciel. Elle peut s'effectuer de diverses manières. La première, sans doute encore la plus courante, est à la main et au ciseau, source fréquente d'énervement. La deuxième, pour les mutants de l'informatique, consiste à réaliser soi-même son dessin en Postscript, merveilleux langage, mais dont la complexité pour le commun des mortels complique l'utilisation et à insérer le programme Postscript dans le source LaTeX. La troisième est fournie par LaTeX lui-même qui met à la disposition de l'utilisateur néophite un environnement «picture » permettant la description de dessins composés de primitives géométriques simples comme des lignes, des cercles ou des rectangles. Cette fonctionnalité de LaTeX est cependant hypothéquée par deux types de contraintes imposées par l'environnement « picture » que nous allons illustrer sur un exemple. Supposons que nous voulions créer un dessin simple comme celui de la figure 1.

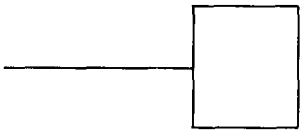

Figure 1 : Un exemple d'utilisation de l'environnement «picture »

Le code LaTeX correspondant s'écrit sous la forme suivante.

```
\begin{picture}(3.8,1.6)(1.3,9.8)
\put(1.28,10.56){\line(1,0){2.45}}
\put(3.74,9.77){\framebox(1.36,1.60){}}
\end{picture}
```

On décrit ici simplement en coordonnées absolues les différentes primitives du dessin. On place donc en (1.28,10.56) une

ligne dont la pente selon x vaut 1 et 0 selon y (donc un segment horizontal). Le calcul de la position des primitives en coordonnées absolues et assez fastidieux et les sources d'erreur assez nombreuses.

La deuxième contrainte est liée à LaTeX lui-même qui impose des régles sévéres sur les primitives. Par exemple les pentes des lignes sont contraintes à être des rapports de nombres premiers entre eux et compris entre ±6. Les primitives se limitent à du texte, des lignes, des cercles, des ovales et des rectangles (figure 2).

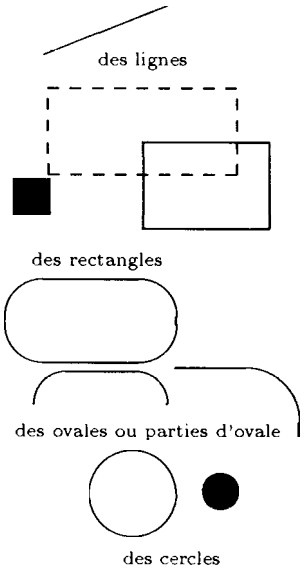

Figure 2 : Les primitives du mode « picture » de LaTeX

JPdraw permet de supprimer la première contrainte et la deuxième peut être surpassée en faisant appel à des primitives Postscript.

2. JPdraw

2.1. Présentation

Après invocation du programme dans l'environnement *suntools* apparaît une fenêtre composée de trois éléments principaux (figure 15) :

- *un tableau d'icône.* Ces icônes permettent de choisir le type de primitives que l'on va créer et d'effectuer quelques manipulations sur celles-ci.
- *un tableau de menus.* Ces menus contiennent les commandes de manipulation des primitives, des fichiers, de mise en page.
- *un canevas* : une surface où apparaît le dessin.

2.2. Création de primitives

Le principe de création est identique à celui de Macdraw. Par exemple pour créer une ligne on clique tout d'abord sur l'icône « ligne » puis en plaçant la souris dans le canevas on clique au point de départ de la ligne et on déplace la souris en maintenant le bouton appuyé jusqu'à ce que la souris occupe la position finale désirée. Cette exemple de création à la souris est intéressant dans la mesure où la pente de la ligne sera rarement compatible avec les régles imposées par LaTeX. Il existe cependant un mode de JPdraw où l'on force les primitives créées à être compatibles avec ces régles. Dans le cas inverse les primitives seront définies par des commandes LaTeX qui feront appel à des programmes Postscript pour générer les figures désirées.

2.3. JPdraw et Postscript

Les régles imposées par LaTeX sont bien sûr incompatibles avec la création des dessins un peu complexes. On peut dépasser ce problème en ajoutant au document LaTeX un fichier de commande (que l'on appelle un *fichier d'extension*) qui va contenir la définition de nouvelle commande LaTeX pour l'environnement « picture » correspondant soit à de nouveaux

types de primitives (par exemple les ellipses) soit à la manipulation des primitives (par exemple la rotation de primitives). Ceci est possible car dans ces nouvelles comma_d_ _ _ f_ir_ _pp_ à des programm_ P _tsc_ipt _ont l_ _ _ _ seront simplement rajoutés au fichier généré après passage du.dvi au .ps. Pour disposer de ces fonctionnalités il est donc nécessaire de disposer d'une imprimante Postscript.

2.4. Les primitives de JPdraw

On dispose, outre les primitives standards de LaTeX, des primitives suivantes (figure 3)

- ellipses
- arcs de cercle
- polygones ou lignes polygonales
- splines, bsplines
- courbes polynomiales
- paraboles
- tracés à la main

On dispose aussi de quatre styles de trait pour chaque primitive : fin, épais, épais pointillé, fin pointillé. On peut de plus régler à sa convenance l'épaisseur des traits épais. Enfin on dispose aussi des tracés de vecteur simple ou double (c'est-à-dire avec des flèches aux deux extrémités).

Remarquons aussi que la création des primitives peut se faire parfois de manière variée. Ainsi, par exemple, les cercles peuvent être crée soit en indiquant le centre puis un point sur le cercle, soit l'inverse, soit en donnant trois points sur le cercle.

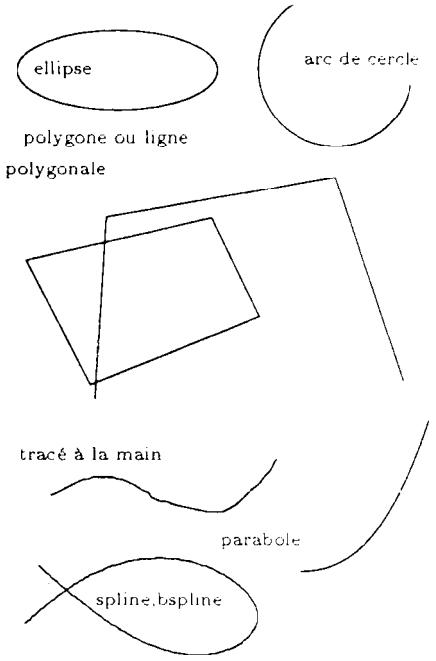

Figure 3 : Les primitives supplémentaires définies dans JPdraw

3. Les manipulations de primitives

3.1. Sélection et association

Il est possible manipuler une primitive après sa création en la *sélectionnant*. Cette opération s'effectue simplement en cliquant à la frontière de la primitive, qui clignote alors pendant un court instant. On peut parfois souhaiter effectuer une manipulation sur un ensemble de primitives, donc en en sélectionnant plusieurs : c'est *l'association*. L'association peut se faire de trois manières :

- association de toutes les primitives contenues dans un rectangle
- association de toutes les primitives successivement sélectionnées
- association des primitives de même type (par exemple tous les textes).

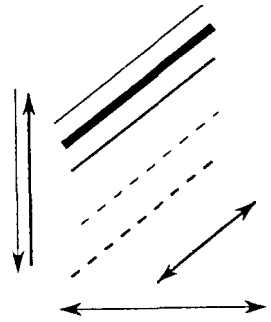

Figure 4 : Les styles de trait de JPdraw

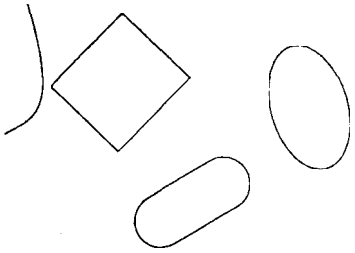

Figure 5 : Exemple de rotation

Toutes les primitives associées subissent en général la même transformation que celle à laquelle une d'entre elles est soumise.

3.2. Les manipulations

Les manipulations décrites dans cette section sont valables aussi bien pour une primitive isolée que pour un ensemble de primitives.

3.2.1. Translation

On peut translater une primitive de trois manières. Tout d'abord à la souris, la primitive sélectionnée suivant les déplacements de la souris, ou bien en indiquant la nouvelle position d'un point donné de la primitive. Enfin, pour les manipulations fines, à l'aide de touche fonction du clavier on peut déplacer les primitives d'un pixel sur l'écran.

3.2.2. Rotation

Toutes les primitives de JPdraw peuvent être soumises à une rotation (y compris les textes), autour d'un point quelconque (figure 5).

3.2.3. Miroirs

On peut transformer une primitive en son image vue dans un miroir (figure 6), le miroir étant placé en diverses positions par rapport à la primitive.

3.2.4. Changement de taille

Les dimensions des primitives peuvent être changées à volonté. Pour les textes les changements de taille consistent soit à modifier le style et la taille des caractères, soit à éditer le texte lui même.

3.2.5. Copier

On peut dupliquer une primitive pour créer un primitive identique à la primitive initiale (éventuellement à un facteur d'homothétie près).

3.2.6. Destruction

On peut détruire une primitive qui reste cependant présente dans les structures de données de JPdraw . Ceci permet par la suite de récupérer une primitive malencontreusement détruite.

3.2.7. Zoom

On peut zoomer le dessin ou une partie du dessin soit dans les deux dimensions, soit selon une direction (figure 7).

4. Quelques utilitaires

4.1. Régles et quadrillages

On peut faire apparaître à l'écran des régles graduées en pixel ou en centimètre ainsi qu'un quadrillage complet de l'écran.

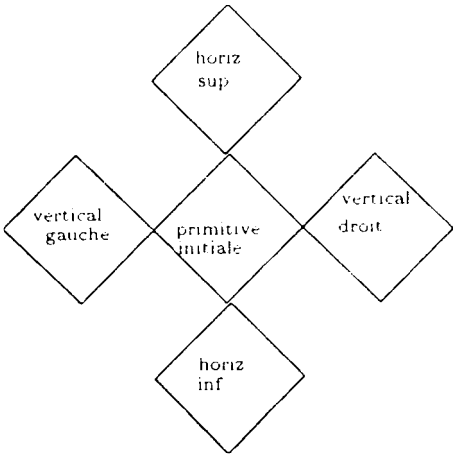

Figure 6 : Exemple de miroir

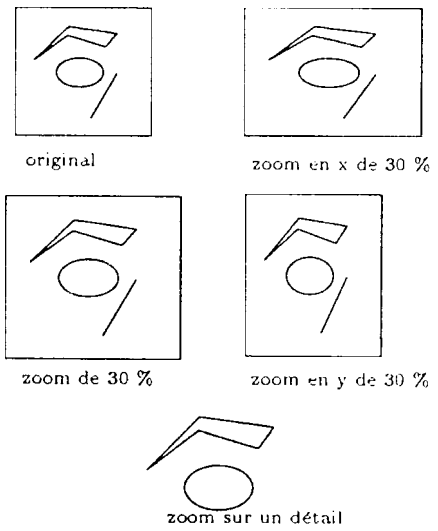

Figure 7 : Exemple de zoom

4.2. Réglures magnétiques

La réglure magnétique est un quadrilllage invisible du canevas qui attire irresistiblement les primitives. Elle permet d'assurer la connection des primitives de façon parfaite. L'alignement des primitives sur la réglures peut se faire a postériori.

5. Textures

Les textures en JPdraw sont de deux natures : soit avec des lignes, soit en niveau de gris.

5.1. Texture de lignes

On peut texturer toute primitive à l'aide de lignes dont on contrôle l'écartement, le style et la pente. On dispose de la possibilité de ne pas texturer des primitives internes à la primitive texturée. La figure 8 montre quelques exemples de textures possibles.

5.2. Texture à niveau de gris

On peut aussi texturer les primitives en niveau de gris. On dispose de 15 niveaux de gris. Une caractéristique intéressante des textures à niveau de gris est qu'elle masque les primitives ou parties de primitive qui sont sous la surface texturée. En utilisant ainsi la texture blanche on peut donc masquer une partie d'une primitive existante. Ainsi sur la figure 9 la partie manquante de l'ellipse a été recouverte par un rectangle que l'on a texturé en blanc.

6. Les fichiers

6.1. Sauver dans un fichier

Les dessins fait en JPdraw sont sauvés sous un format interne dans un fichier dont l'extension est .macdraw. La commande *sauver* place dans un fichier dont le nom est donné par l'utilisateur (par défaut xx.ltex) l'environnement « picture » décrivant le dessin. Pour inclure le dessin dans un source LaTeX il suffit de placer un \input{xx.ltex} dans le source LaTeX. Une possibilité intéressante pour les gros dessins est de sauver le dessin en partie en Postscript (tout sauf les textes) et en par-

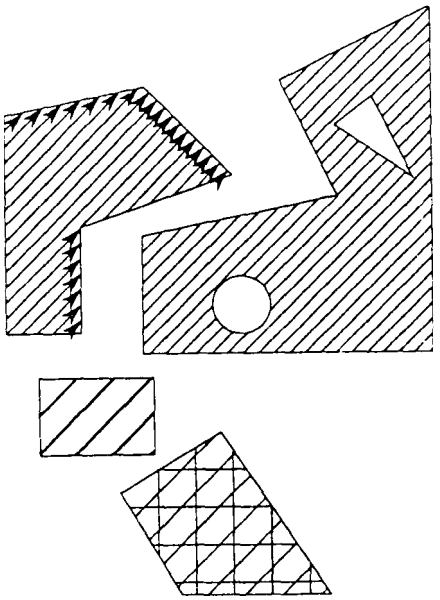

Figure 8 : Exemple de texture à base de ligne

tie en LaTeX (les textes). Ceci permet de garder la puissance de LaTeX pour le traitement du texte (par exemple pour les mathématiques) et d'avoir le dessin en postscript. Dans ce cas JPdraw génére un troisième fichier d'extension .ps qui contient la description du dessin. Par exemple pour le dessin 3 sauvé dans ce mode JPdraw génére un fichier LaTeX qui est :

```
\begin{picture}(5.6,8.2)(0.2,5.1)
\scriptsize
\put(0.57,12.75){ellipse}
\put(0.42,11.85){polygone ou ligne }
\put(0.19,11.43){polygonale}
\put(0.38,7.66){trac\'e \'a la main }
\put(1.38,5.74){spline,bspline}
\put(3.00,6.54){parabole}
\put(0.2, 5.1)
{\special{ps:plotfile fig3.ps}}
\end{picture}
```

Le fichier fig3.ps contient simplement le dessin. Pour l'utilisateur le mode d'inclusion du dessin dans le source LaTeX reste transparent.

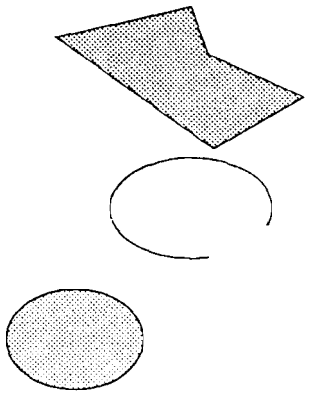

Figure 9 : Exemple de texture à niveau de gris

6.2. Charger et Album

JPdraw ne peut recharger que des fichiers au format .sacdraw. On peut donner un nom de fichier au lancement du programme ou bien utiliser la commande *charger* de JPdraw . La commande *album* permet de charger un dessin sans détruire le dessin courant.

7. Le méta-language de JPdraw

Dans les sections précédentes nous venons de décrire les possibilités de JPdraw sur des primitives qui sont crées à la souris. Cependant on pourrait désirer créer ces primitives non plus à la souris mais à l'aide de commandes décrites dans un fichier. Par exemple un expérimentateur ayant dans un fichier les coordonnées des points d'une fonction pourrait désirer créer le tracé de sa fonction en utilisant directement ce fichier sans reprendre à la main les points de la fonction. Ceci est possible en utilisant le méta-language de JPdraw .

Ce language est basé sur la notion de mode qui décrit soit une primitive soit une manière de connecter des points. Par

exemple le *mode connecté* décrit le fait que tous les points définis dans la suite du fichier doivent être connectés par des segments de droites. C'est ce mode qui a été utilisé pour le tracé de fonction de la figure 13. Dans le fichier initial nous avions simplement placé le texte suivant : mode connecte suivi des coordonnées x,y de la fonction. JPdraw génère alors automatiquement le tracé de la fonction dans le rectangle défini par l'utilisateur. En option on peut postérieurement tracer automatiquement les échelles comme cela a été fait dans cette figure. Donnons maintenant un exemple simple d'utilisation basé sur l'utilisation d'un système de CAO. On a modélisé un cube avec ce système qui écrit à la demande un fichier dans le méta-language de JPdraw . On peut alors récupérer une vue 3D de ce cube en LaTeX. Ici on utilise le *mode deconnecte* où les paires de points indiquent les coordonnées de début et de fin d'un segment. Le fichier en méta-language s'écrit :

```
mode deconnecte
    -0.00683    -0.00532
     0.00183    -0.00703
     0.00183    -0.00703
     0.00183     0.00236
     0.00183     0.00236
    -0.00683     0.00407
    -0.00683     0.00407
    -0.00683    -0.00532
     0.00183    -0.00703
     0.00683    -0.00407
     0.00683    -0.00407
     0.00683     0.00532
     0.00683     0.00532
     0.00183     0.00236
     0.00683     0.00532
    -0.00183     0.00703
    -0.00683     0.00407
    -0.00183     0.00703
```

et permet de créer le dessin de la figure 10.

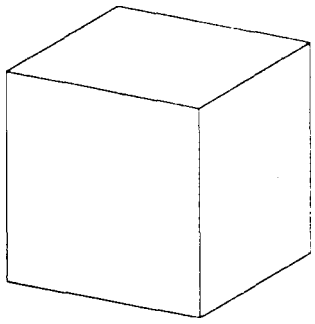

Figure 10 : Un exemple d'utilisation du méta-language de JPdraw

8. Autour de JPdraw

Différent utilitaires gravitent autour de JPdraw permettant la conversion d'un environnement « picture » en format macdraw, la conversion (partielle) d'un fichier postscript au format macdraw ou bien la création d'un code exécutable par un plotter de type Hewlett-Packard à partir d'un dessin JPdraw pour la création de transparents couleurs.

9. Quelques exemples

On donne dans cette section quelques exemples de dessins crées soit à la souris soit à l'aide du méta-language.

9.1. CAO

Voici un exemple d'utilisation d'un système de CAO interfacé avec JPdraw . Le texte a été rajouté à la main

9.2. Satellite

Ce dessin a été crée à la souris. On a tout d'abord crée la Terre et une orbite avec le cône d'observation du satellite. Ces deux élément ont été copiés, associés, puis soumis à une rotation autour du centre de la Terre. Des polygones texturés ont

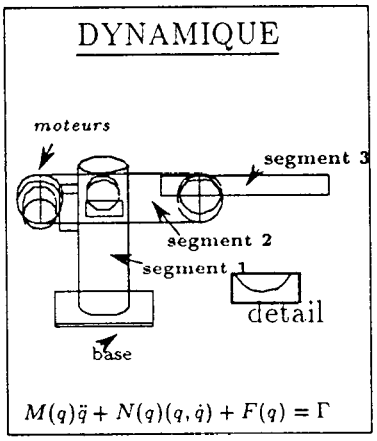

Figure 11 : CAO

permis ensuite de créer les continents et de masquer les parties d'orbite derrière la Terre.

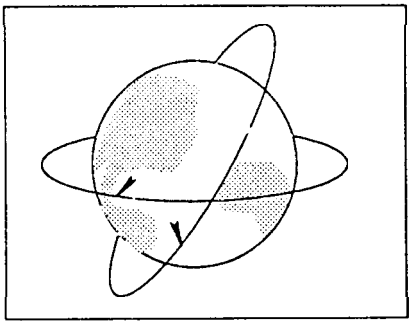

Figure 12 : Satellite

9.3. Tracé de fonction

On a simplement utilisé les résultats d'une expérience de robotique qui avait permis la création d'un fichier de points de la fonction. Le quadrillage est placé automatiquement par JPdraw .

9.4. Une pince

Voici un exemple d'utilisation des polygones avec des textures à niveau de gris. Le modèle est issu directement d'un système de CAO.

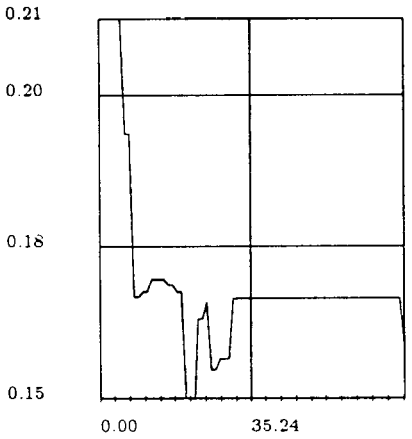

Figure 13 : Tracé de fonction

10. Conclusion

JPdraw est un outil sans ambition qui permet la création de manière relativement simple de dessin dans l'environnement LaTeX. Son développement a été assuré pour faciliter la création immédiate de dessins à partir de résultats expérimentaux. L'exécutable est disponible librement (sous réserve d'une mise à disposition pour les utilisateurs non-universitaires).

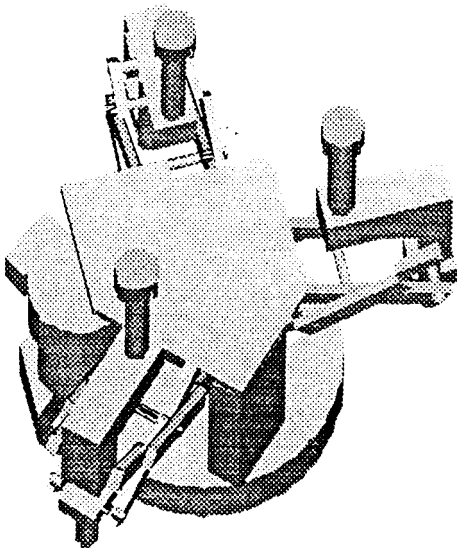

Figure 14: La pince

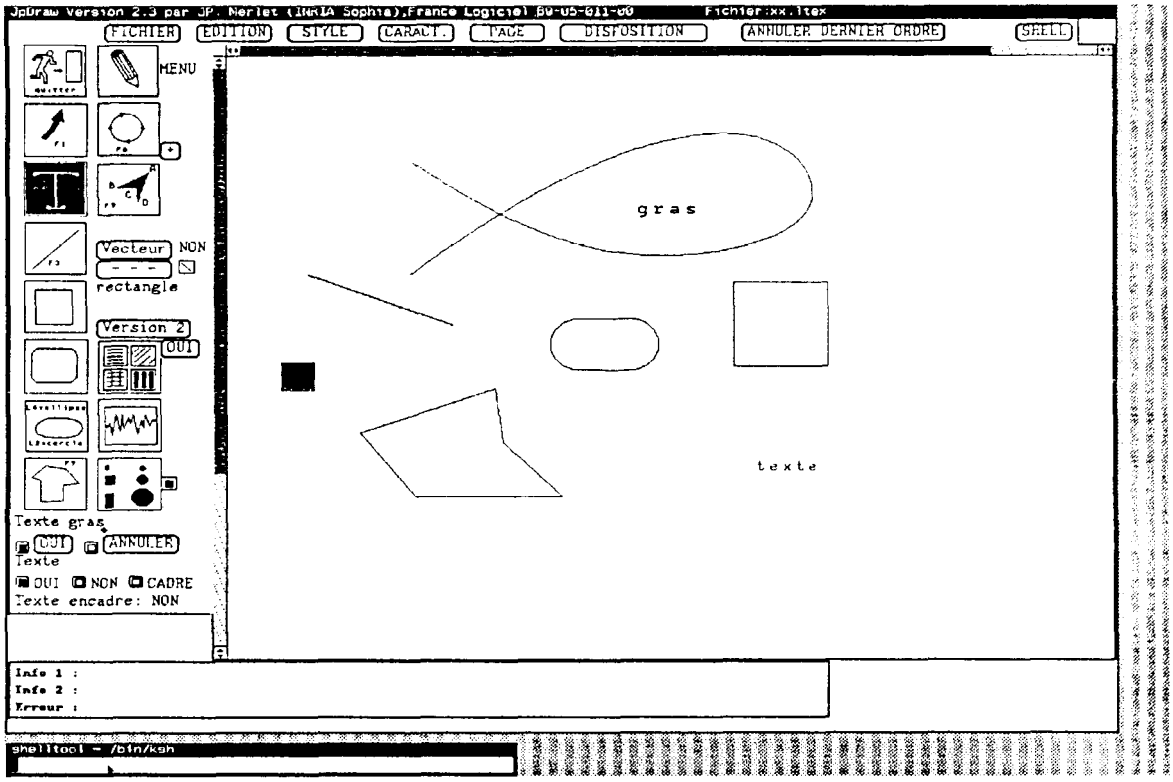

Figure 15 : La fenêtre JPdraw