

Extension 3d pour le format jps

par Philippe Saadé
(psaade.jps@free.fr)
22 Décembre 2005

1. Général

$x y z$ **SetCamPos** - \rightarrow Positionne la caméra au point (x, y, z)

- **GetCamPos** $x y z$ \rightarrow Dépose sur la pile les coordonnées de la caméra

$V_x V_y V_z$ **SetCamVec** - \rightarrow Set Camera Looking vector.

- **GetCamVec** $V_x V_y V_z$ \rightarrow Get Camera Looking vector.

$U_x U_y U_z$ **SetCamUp** - \rightarrow Set Camera Up vector.

- **GetCamUp** $U_x U_y U_z$ \rightarrow Set Camera Up vector.

- **ComputeCamera** - \rightarrow Compute vectors usefull to CamView.

$x y z$ **CamView** $X Y$ \rightarrow On projete le point 3d sur le plan de représentation de la caméra, selon le mode de représentation

$array1$ **tab3dto2d** $array2$ \rightarrow transforme un tableau de points 3d en tableau de points 2d

- **qplanxy** - \rightarrow Trace un quadrillage du plan XY

getp3d \rightarrow

- *representationtype* : Chaîne de caractère spécifiant le type de perspective : (perspective) ou (ortho). **valeur par défaut** : (perspective)
- *ScreenDist* : Distance par rapport à l'écran. **valeur par défaut** : 0.1
- *ZoomFactor_x* : Facteur de zoom en x . **valeur par défaut** : 100
- *ZoomFactor_y* : Facteur de zoom en y . **valeur par défaut** : 100

2. Les axes et quadrillages

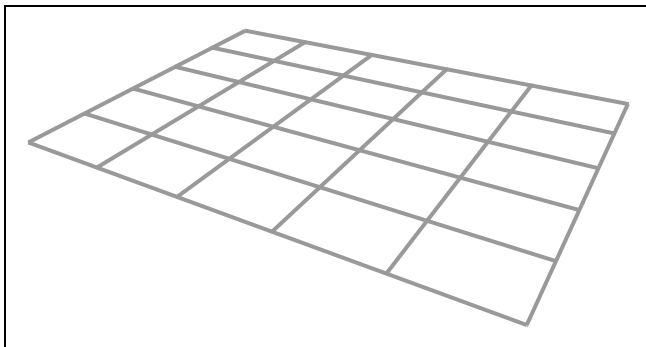
$xmin xmax ymin ymax zmin zmax$ **quadrilleXYZ** - \rightarrow Effectue un quadrillage d'unité 1 sur le produit $[xmin; xmax] \times [ymin; ymax] \times [zmin; zmax]$

$xmin xmax \ell$ **axeR** - \rightarrow $[xmin; xmax]$ = étendue du pointille, ℓ = longueur du vecteur

$ymin ymax \ell$ **axeV** - \rightarrow $[ymin; ymax]$ = étendue du pointille, ℓ = longueur du vecteur

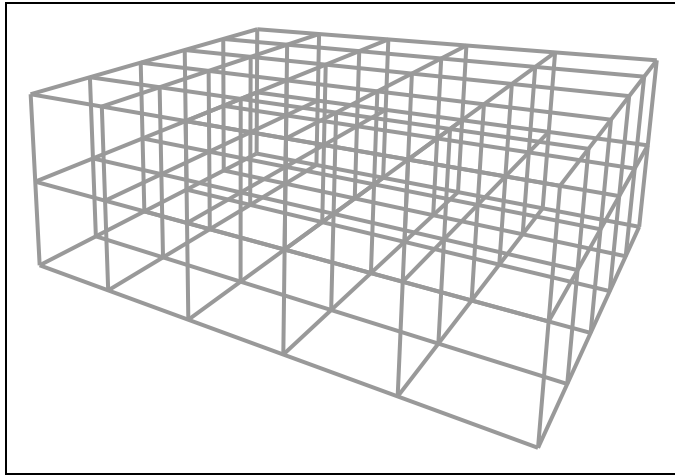
$zmin zmax \ell$ **axeB** - \rightarrow $[zmin; zmax]$ = étendue du pointille, ℓ = longueur du vecteur

$min max \ell$ **axeRVB** - \rightarrow $[min; max]$ = étendue des pointillés, ℓ = longueur des vecteurs



source jps

```
autocrop
%% échelle et étendue de l'image
30 setxunit
-1 9 setxrange
-5 2 setyrange
%% Positionnement de la Caméra
6 -6 4 SetCamPos
-1 1.1 -0.3 SetCamVec
0.03 0.03 1 SetCamUp
ComputeCamera
%% traces
1.5 setlinewidth
.6 setgray
0 5 0 5 0 0 quadrilleXYZ
```

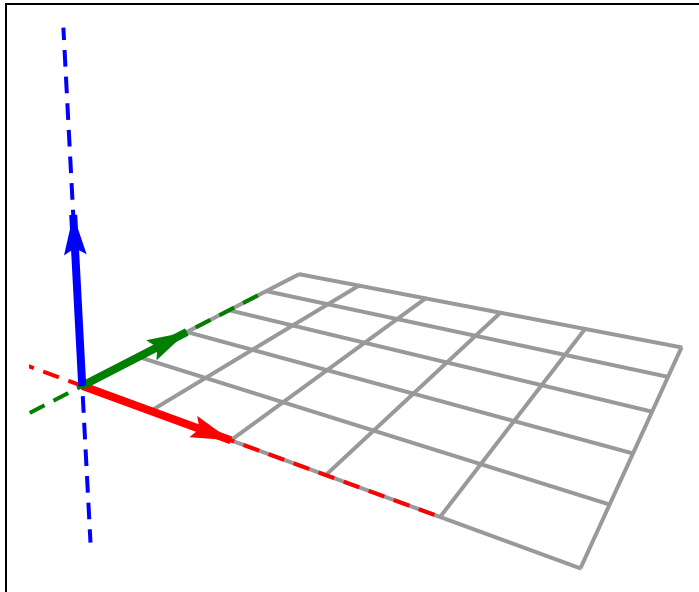


source jps

```

autocrop
%% échelle et étendue de l'image
30 setxunit
-1 9 setxrange
-5 2 setyrange
%% Positionnement de la Caméra
6 -6 4 SetCamPos
-1 1.1 -0.3 SetCamVec
0.03 0.03 1 SetCamUp
ComputeCamera
%% traces
1.5 setlinewidth
.6 setgray
0 5 0 5 0 2 quadrilleXYZ

```



source jps

```

autocrop
%% échelle et étendue de l'image
30 setxunit
-1 9 setxrange
-5 2 setyrange
%% Positionnement de la Caméra
6 -6 4 SetCamPos
-1 1.1 -0.3 SetCamVec
0.03 0.03 1 SetCamUp
ComputeCamera
%% traces
1.5 setlinewidth
.6 setgray
0 5 0 5 0 0 quadrilleXYZ
3 setlinewidth
/arrowscale {2 2} def
-2 4 2 axesRVB

```

3. Opérateurs

3.1 - Sur les points

$A B$ **distance3d** d \rightarrow calcule la distance $d = AB$

A **dupp3d** $A A$ \rightarrow Duplique le point 3d au dessus de la pile

$x y z$ *lit* **defpoint3d** $-$ \rightarrow Associe le littéral *lit* au point (x, y, z)

$M A \vec{v}$ **orthoprojplane3d** M' \rightarrow Le point M' est le projeté du point M sur le plan P défini par le point A et le vecteur \vec{v} , normal à P .

$A B$ **milieu3d** I $\rightarrow I$ est le milieu de $[AB]$

3.2 - Sur les vecteurs

$A B$ **vecteur3d** u $\rightarrow u = \overrightarrow{AB}$

\vec{u} **norme3d** r $\rightarrow r$ est la norme du vecteur \vec{u}

u **dupv3d** $u u$ \rightarrow Duplique le vecteur u au dessus de la pile

$\vec{u} \vec{v}$ **addv3d** \vec{w} $\rightarrow \vec{w} = \vec{u} + \vec{v}$

$\vec{u} \lambda$ **mulv3d** \vec{v} $\rightarrow \vec{v} = \lambda \vec{u}$

$\lambda \vec{u}$ **lambdav3d** \vec{v} \rightarrow Le vecteur \vec{v} vérifie $\vec{v} = \lambda \vec{u}$

$\vec{u} \vec{v}$ **vectprod3d** \vec{w} $\rightarrow \vec{w} = \vec{u} \wedge \vec{v}$

$\vec{u} \vec{v}$ **subv3d** \vec{w} $\rightarrow \vec{w} = \vec{u} - \vec{v}$

\vec{u} **unitaire3d** \vec{v} \rightarrow Si $\vec{u} = \vec{0}$, alors $\vec{v} = \vec{0}$, sinon $\vec{v} = \vec{u} / \|\vec{u}\|$

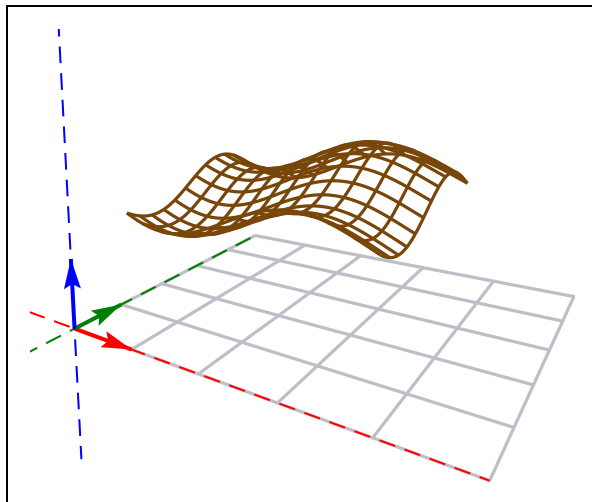
\vec{u} **normalize3d** \vec{v} \rightarrow Synonyme de **unitaire3d** : si $\vec{u} = \vec{0}$, alors $\vec{v} = \vec{0}$, sinon $\vec{v} = \vec{u}/\|\vec{u}\|$
 \vec{u} \vec{v} **scalprod3d** s \rightarrow Produit scalaire : $s = \vec{u} \cdot \vec{v}$

4. Commandes de tracés

A **plus3d** \rightarrow Analogue 3d de la commande **plus**
A **point3d** \rightarrow Analogue 3d de la commande **point**
array **points3d** \rightarrow Analogue 3d de la commande **points**
array **ligne3d** \rightarrow Analogue 3d de la commande **ligne**
array **polygone3d** \rightarrow Analogue 3d de la commande **polygone**
array **polygone*3d** \rightarrow Analogue 3d de la commande **polygone***

5. Surfaces

$xmin$ pas_x $xmax$ $ymin$ pas_y $ymax$ f **surfaceparam3d** \rightarrow Dessine la surface $f(x,y) = z$ sur $[xmin; xmax] \times [ymin; ymax]$. f est un exécutable.



source *jps*

```

autocrop
%% échelle et étendue de l'image
25 setxunit
-1 9 setxrange
-5 2 setyrange
%% Positionnement de la Caméra
6 -6 4 SetCamPos
-1 1.1 -0.3 SetCamVec
0.03 0.03 1 SetCamUp
ComputeCamera
%% traces
1.2 setlinewidth
190 255 div
190 255 div
200 255 div setrgbcolor
0 5 0 5 0 0 quadrilleXYZ

1.5 setlinewidth
/arrowscale {1.5 dup} def
-2 5 1 axesRVB

%% la fonction z = f (x, y)
/f { % x y
2 dict begin
  /y exch def
  /x exch def
#rpn# Cos (x-y-1)* 0.5 * Cos (x+y+1) + 2
end
} def

1.2 setlinewidth
120 255 div
70 255 div
9 255 div setrgbcolor

/pas 0.25 def
1 pas 4 0 pas 3 {f} surfaceparam3d

```

6. Placement de texte ou de labels \TeX

ultext3d - → Analogue 3d de la commande **ultext**
dctext3d - → Analogue 3d de la commande **dctext**
crtext3d - → Analogue 3d de la commande **crtext**
cltext3d - → Analogue 3d de la commande **cltext**
bctext3d - → Analogue 3d de la commande **bctext**
dbtext3d - → Analogue 3d de la commande **dbtext**
ultexlabel3d - → Analogue 3d de la commande **ultexlabel**
bctexlabel3d - → Analogue 3d de la commande **bctexlabel**
bbtext3d - → Analogue 3d de la commande **bbtext**
cctexlabel3d - → Analogue 3d de la commande **cctexlabel**
dctexlabel3d - → Analogue 3d de la commande **dctexlabel**
urtextlabel3d - → Analogue 3d de la commande **urtextlabel**
dltext3d - → Analogue 3d de la commande **dltext**
uctext3d - → Analogue 3d de la commande **uctext**
bltexlabel3d - → Analogue 3d de la commande **bltexlabel**
cctext3d - → Analogue 3d de la commande **cctext**
cltexlabel3d - → Analogue 3d de la commande **cltexlabel**
brtext3d - → Analogue 3d de la commande **brtext**
ubtexlabel3d - → Analogue 3d de la commande **ubtexlabel**
bltext3d - → Analogue 3d de la commande **bltext**
dltexlabel3d - → Analogue 3d de la commande **dltexlabel**
ubtext3d - → Analogue 3d de la commande **ubtext**
cbtext3d - → Analogue 3d de la commande **cbtext**
brtexlabel3d - → Analogue 3d de la commande **brtexlabel**
crtexlabel3d - → Analogue 3d de la commande **crtexlabel**
bbtexlabel3d - → Analogue 3d de la commande **bbtexlabel**
cbtexlabel3d - → Analogue 3d de la commande **cbtexlabel**
uctexlabel3d - → Analogue 3d de la commande **uctexlabel**
dbtexlabel3d - → Analogue 3d de la commande **dbtexlabel**
urtext3d - → Analogue 3d de la commande **urtext**