

Macros shadow . mp

CHUPIN Maxime mc@melusine.eu.org

Table des matières

1	Introduction	1
2	Notes	1
3	Ombre « classique »	1
4	Ombre « dégradée »	3
5	Ombre « blur »	3
6	Le code	5

1 Introduction

Je me suis aperçu qu'il manquait la gestion de l'ombre dans METAPOST, et même dans Metafun. Alors, j'ai essayé de faire quelques macros permettant de réaliser plusieurs types d'ombres assez simplement. Ces fonctions sont, je pense, loins d'être optimales. Libre à ceux qui veulent les améliorer de le faire.

2 Notes

Remarque

Attention, ces macros incluent Metafun, vous devez donc en disposer : www.pragma-ade.com/general/manuals/. De plus, le format .eps ne permet pas la visualisation des dégradés. Dans ce document, j'ai donc été obligé d'utiliser pdfL^AT_EX.

Dans tous les exemples suivants, les codes sources METAPOST commenceront par :

Code :

```
verbatimtex
%&latex
\documentclass{article}
\usepackage[utf8]{inputenc}
\usepackage[charter]{mathdesign}
\begin{document}
etex
```

Je ne remettrai pas ce « préambule » dans un soucis de lisibilité.

3 Ombre « classique »

Le premier type d'ombre est l'ombre que je qualifie de « classique », c'est-à-dire la reproduction de l'image simplement décalée.

Ceci s'obtient simplement avec la commande `shadow(<path ou picture>, <pair>, <couleur>)`, où le path (ou picture) est ce à quoi on veut appliquer l'ombre, le pair représente le vecteur de décalage de l'ombre, et la couleur est la couleur de l'ombre.

Voici un exemple :

 Code :

```
input shadow;

beginfig(1);
  shadowrect(btex Essai etex scaled 16, (3,-3),black);
  draw btex Essai etex scaled 16 withcolor blue withpen pencircle scaled 10pt;
endfig;
```

Ceci produit :



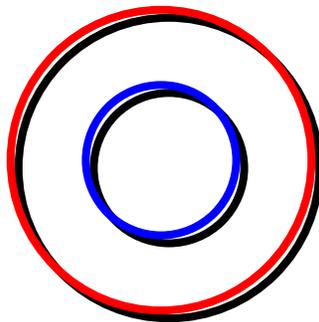
Dans l'exemple suivant, on trace l'ombre d'une picture.

 Code :

```
input shadow;

beginfig(0);
  picture im;
  im:=image(
    draw fullcircle scaled 4cm withpen pencircle scaled 3pt withcolor red;
    draw fullcircle scaled 2cm withpen pencircle scaled 3pt withcolor blue;
  );
  shadowrect(im, (3,-3), black);
  draw im;
endfig;
```

Ce qui donne :



4 Ombre « dégradée »

Voici maintenant le deuxième style d'ombre, qui n'est en fait pas vraiment une ombre mais ceci produit un effet qui peut être intéressant. Le principe est de répéter l'image dont on veut produire l'ombre le long du vecteur donné en argument avec une couleur de plus en plus transparente.

La syntax est `shadowdeg(<picture ou path>,<pair>,<color>,<précision>)`, où comme précédemment, le path (ou picture) est ce à quoi on veut appliquer l'ombre, le pair est le vecteur qui indique le décalage de l'ombre, color est la couleur de l'ombre et la précision est le nombre d'images répétées le long du vecteur. Voici deux exemples illustrant ce qui vient d'être expliqué :

 Code :

```
beginfig(2);
  shadowdeg(btex Essai etex scaled 16, (4,-4),black,200);
  draw btex Essai etex scaled 16 withcolor blue withpen pencircle scaled 10pt;
endfig;
```

qui donne :



et,

 Code :

```
beginfig(3);
  shadowdeg(btex Essai etex scaled 16, (6,-6),black,4);
  draw btex Essai etex scaled 16 withcolor blue withpen pencircle scaled 10pt;
endfig;
```

qui donne :



5 Ombre « blur »

Voici le dernier type d'ombre, l'ombre « blur ». Le principe est d'avoir une ombre du type de l'ombre « classique » mais avec un effet de dispersion tout au tour.

La syntax est `shadowblur(<picture ou path>,<pair>,<précision>,<blur>)`. Là aussi, le path (ou picture) est ce à quoi on veut appliquer l'ombre et le pair est le vecteur de distance de l'ombre. La précision et le blur sont deux numeric, ils règlent la « dispersion » et sont liés, blur est relié à la distance de dispersion autour de l'ombre « classique » et précision est le nombre de passes pour réaliser cette dispersion. Attention ce procédé est **très** lourd car la macro génère beaucoup d'images.

Voici trois exemples :

 Code :

```
beginfig(4);  
  shadowblur(btex Essai etex scaled 16, (3,-3),black,50,4);  
  draw btex Essai etex scaled 16 withcolor blue withpen pencircle scaled 10pt;  
endfig;
```

qui donne :



 Code :

```
beginfig(5);  
  shadowblur(btex Essai etex scaled 16, (3,-3),black,50,12);  
  draw btex Essai etex scaled 16 withcolor blue withpen pencircle scaled 10pt;  
endfig;
```



 Code :

```
beginfig(6);  
  shadowblur(btex Essai etex scaled 16, (3,-3),black,3,12);  
  draw btex Essai etex scaled 16 withcolor blue withpen pencircle scaled 10pt;  
endfig;
```

Essai

6 Le code

 Code :

```
1  input metafun;
2
3  vardef shadowrect(expr dessin, vect, couleur) =
4      save im,i;
5      picture im;
6      im := image(
7          draw dessin;
8      );
9      draw im shifted vect withcolor couleur;
10 enddef;
11
12 vardef shadowblur(expr dessin, vect, couleur,pre,blur) =
13     save im,i,t;
14     picture im;
15     im := image(
16         draw dessin;
17     );
18     draw im shifted vect withcolor couleur;
19     for i:=pre downto 0:
20         for t:= 0 step 10 until 360:
21             draw im shifted (vect+blur*i/pre*(sind(t),cosd(t))) withcolor transparent
22                 (1,0.5,((0.05,0.05,0.05)+couleur+(i)/pre*white));
23         endfor;
24     endfor;
25 enddef;
26
27 vardef shadowdeg(expr dessin, vect, couleur,pre) =
28     save im,i;
29     picture im;
30     im := image(
31         draw dessin;
32     );
33     for i:=pre downto 0:
34         draw im shifted ((i/pre)*vect) withcolor transparent(1,(10/(pre))*(pre-i)/pre),couleur
35     );
36     endfor;
37 enddef;
```