

man mlx_new_image

MiniLibX 3 " 19 septembre 2002 ".

NOM MiniLibX - Manipulation d'images

SYNOPSIS

```
void * mlx_new_image(void *mlx_ptr, int width, int height) ;  
char *mlx_get_data_addr(void *img_ptr, int *bits_per_pixel, int *size_line, int *endian) ;  
int mlx_put_image_to_window(void *mlx_ptr, void *win_ptr, void *img_ptr, int x, int y) ;  
unsigned int mlx_get_color_value(void *mlx_ptr, int color) ;  
void *mlx_xpm_to_image(void *mlx_ptr, char **xpm_data, int *width, int *height) ;  
void *mlx_xpm_file_to_image(void *mlx_ptr, char *filename, int *width, int *height) ;  
void *mlx_png_file_to_image(void *mlx_ptr, char *filename, int *width, int *height) ;  
int mlx_destroy_image(void *mlx_ptr, void *img_ptr) ;
```

DESCRIPTION

[mlx_new_image\(\)](#) crée une nouvelle image en mémoire. Elle renvoie un identifiant void *

nécessaire à la manipulation ultérieure de cette image. Elle n'a besoin que de la taille de l'image à créer, en utilisant les paramètres width et height, et de l'identifiant de connexion mlx_ptr (voir le manuel mlx).

L'utilisateur peut dessiner à l'intérieur de l'image (voir ci-dessous), et peut vider l'image dans une fenêtre spécifiée à tout moment pour l'afficher à l'écran. Ceci est fait en utilisant [mlx_put_image_to_window\(\)](#). Trois identifiants sont nécessaires ici, pour la connexion à l'écran, la fenêtre à utiliser, et l'image (respectivement mlx_ptr, win_ptr et img_ptr). Les coordonnées (x, y) définissent où l'image doit être placée dans la fenêtre.

[mlx_get_data_addr\(\)](#) renvoie des informations sur l'image créée, permettant à l'utilisateur de la modifier ultérieurement. Le paramètre img_ptr spécifie l'image à utiliser. Les trois paramètres suivants doivent être les adresses de trois entiers valides différents.

Le paramètre bits_per_pixel sera rempli avec le nombre de bits nécessaires pour représenter la couleur d'un pixel (également appelé la profondeur de l'image).

size_line est le nombre d'octets utilisés pour stocker une ligne de l'image en mémoire. Cette information est nécessaire pour se déplacer d'une ligne à l'autre dans l'image.

endian indique si la couleur des pixels de l'image doit être stockée en little endian (endian == 0), ou en big endian (endian == 1).

mlx_get_data_addr renvoie une adresse char * qui représente le début de la zone mémoire où est stockée l'image. A partir de cette adresse, les premiers bits_par_pixel représentent la couleur du premier pixel de la première ligne de l'image. Le deuxième groupe de bits_par_pixel représente le deuxième pixel de la première ligne, et ainsi de suite.

Ajoutez size_line à l'adresse pour obtenir le début de la deuxième ligne. Vous pouvez atteindre n'importe quel pixel de l'image de cette manière.

mlx_destroy_image détruit l'image donnée (img_ptr).

STOCKAGE DE LA COULEUR DANS LES IMAGES

Selon l'affichage, le nombre de bits utilisés pour stocker la couleur d'un pixel peut changer. L'utilisateur représente généralement une couleur en mode RVB, en utilisant

un octet pour chaque composante (voir le manuel mlx_pixel_put). Ceci doit être traduit pour s'adapter à l'exigence de bits_par_pixel de l'image, et rendre la couleur compréhensible pour l'affichage.

C'est le but de la fonction mlx_get_color_value(). Elle prend un paramètre de couleur RVB standard, et renvoie une valeur int non signée.

Les bits_par_pixel les moins significatifs de cette valeur peuvent être stockés dans l'image. Vous pouvez éviter d'utiliser cette fonction si aucune conversion n'est nécessaire (par exemple, en cas de profondeur de 24 bits ou de 32 bits).

Gardez à l'esprit que la position des bits les moins significatifs dépend de l'endian de l'ordinateur local.

l'endian de l'ordinateur local. Si l'endian de l'image diffère de l'endian local (ce qui ne devrait se produire que dans un environnement réseau X11), alors la valeur doit être transformée avant d'être utilisée.

IMAGES XPM ET PNG

Les fonctions mlx_xpm_to_image() , mlx_xpm_file_to_image() et mlx_png_file_to_image () créent une nouvelle image de la même manière. Elles la rempliront en utilisant les données xpm_data ou le nom de fichier spécifié, selon la fonction utilisée.

Notez que MiniLibX n'utilise pas les bibliothèques standard Xpm et png pour traiter les

images xpm et png. Il se peut que vous ne puissiez pas lire tous les types d'images xpm et png. Elle gère cependant la transparence.

VALEURS RETOURNÉES

Les quatre fonctions qui créent des images, `mlx_new_image()`, `mlx_xpm_to_image()`, `mlx_xpm_file_to_image()` et `mlx_png_file_to_image()`, renverront NULL si une erreur se produit. Sinon, elles renvoient un pointeur non nul comme identifiant d'image.

VOIR AUSSI

`mlx(3)`, `mlx_new_window(3)`, `mlx_pixel_put(3)`, `mlx_loop(3)`