

**NAME**

MiniLibX - Drawing inside windows

**SYNOPSIS**

```
int
mlx_pixel_put ( void *mlx_ptr, void *win_ptr, int x, int y, int color );

int
mlx_string_put ( void *mlx_ptr, void *win_ptr, int x, int y, int color, char *string );
```

**DESCRIPTION**

The **mlx\_pixel\_put** () function draws a defined pixel in the window *win\_ptr* using the ( *x* , *y* ) coordinates, and the specified *color* . The origin (0,0) is the upper left corner of the window, the x and y axis respectively pointing right and down. The connection identifier, *mlx\_ptr* , is needed (see the **mlx** man page).

Parameters for **mlx\_string\_put** () have the same meaning. Instead of a simple pixel, the specified *string* will be displayed at ( *x* , *y* ).

In both functions, it is impossible to display anything outside the specified window, nor display in another window in front of the selected one.

**COLOR MANAGEMENT**

The *color* parameter has an integer type. The displayed color needs to be encoded in this integer, following a defined scheme. All displayable colors can be split in 3 basic colors: red, green and blue. Three associated values, in the 0-255 range, represent how much of each color is mixed up to create the original color. These three values must be set inside the integer to display the right color. The three least significant bytes of this integer are filled as shown in the picture below:

```
| 0 | R | G | B |  color integer
+---+---+---+---+
```

While filling the integer, make sure you avoid endian problems. Remember that the "blue" byte should always be the least significant one.

**SEE ALSO**

mlx(3), mlx\_new\_window(3), mlx\_new\_image(3), mlx\_loop(3)

**AUTHOR**

Copyright ol@ - 2002-2015 - Olivier Crouzet