

НАЗВАНИЕ

MiniLibX - обработка событий

СИНОПСИС

```
int    mlx_loop ( void *mlx_ptr );  
  
int    mlx_key_hook ( void *win_ptr, int (*funct_ptr)(), void *param );  
  
int    mlx_mouse_hook ( void *win_ptr, int (*funct_ptr)(), void *param );  
  
int    mlx_expose_hook ( void *win_ptr, int (*funct_ptr)(), void *param );  
  
int    mlx_loop_hook ( void *mlx_ptr, int (*funct_ptr)(), void *param );
```

X-WINDOW СОБЫТИЯ

Система X-Window является двунаправленной. С одной стороны, программа отправляет заказы на экран для отображения пикселей, изображений и т.д.

С другой стороны, он может получать информацию от клавиатуры и мыши, связанных с экраном. Для этого программа получает «события» от клавиатуры или мыши.

ОПИСАНИЕ

Чтобы получать события, вы должны использовать `mlx_loop()`. Эта функция никогда не возвращается.

Это бесконечный цикл, который ожидает события, а затем вызывает пользовательскую функцию, связанную с этим событием. Необходим один параметр - идентификатор соединения `mlx_ptr` (см. Руководство по `mlx`). Вы можете назначить разные функции трем следующим событиям:

- Клавиша нажата
- кнопка мыши нажата
- Апартамент окна должен быть перерисован (это называется событием «выставить», и задача вашей программы - обработать его).

Каждое окно может определять разные функции для одного и того же события.

Три функции `mlx_key_hook()`, `mlx_mouse_hook()` и `mlx_expose_hook()` работают точно так же.

`funct_ptr` - указатель на функцию, которую вы хотите вызвать, когда происходит событие.

Это назначение относится к окну, определенному идентификатором win_ptr.

Адрес param будет передаваться функции каждый раз, когда она вызывается, и должен использоваться для хранения параметров, которые могут ему понадобиться.

Синтаксис для функции mlx_loop_hook() идентичен предыдущим, но данная функция будет вызываться, когда событие не происходит. Когда она ловит событие, MiniLibX вызывает соответствующую функцию с фиксированными параметрами:

```
expose_hook(void *param);  
  
key_hook(int keycode,void *param);  
  
mouse_hook(int button,int x,int y,void *param);  
  
loop_hook(void *param);
```

Эти имена функций являются произвольными. Они здесь используются для различения параметров в зависимости от события. Эти функции НЕ являются частью MiniLibX.

param - адрес, указанный в вызовах mlx_*_hook. Этот адрес никогда не используется и не изменяется MiniLibX. При событиях, связанных с клавишами и мышью, передается дополнительная информация: код клавиши сообщает, какая клавиша нажата (ищите включаемый файл X11 «keysymdef.h»), (x,y) - координаты щелчка мышью в окне и button говорит вам, какая кнопка мыши была нажата.

ДАЛЕЕ С СОБЫТИЯМИ

MiniLibX обеспечивает общий доступ ко всем событиям X-Window.

Mlx.h включает определение mlx_hook() таким же образом, как работают функции mlx_*_hook.

Значения события и маски будут взяты из включаемого файла X11 «X.h».

Посмотрите исходный код mlx_int_param_event.c, чтобы узнать, как MiniLibX будет вызывать вашу собственную функцию для определенного события.