

Lazy Foo' Productions

[News](#) [FAQs](#) [Games](#) [Tutorials](#) [Articles](#) [Contact](#) [Donations](#)

Interface Explorer - HL7

Tool to easily read/edit messages and debug interfaces: 45 day eval.

Quantum Research Group

Leader in capacitive touch sensors. Buttons, wheels, sliders & screens.

Ad by Google

Key States



Last Updated 11/15/07

Here you're going to check if a key is pressed without even using events. This is possible due to keystates. Sometimes just checking if a key is up or down is easier than monitoring events.

This tutorial will teach you to check if the key is pressed, rather than monitoring key events.

```
//While the user hasn't quit
while( quit == false )
{
    //While there's events to handle
    while( SDL_PollEvent( &event ) )
    {
        //If the user has Xed out the window
        if( event.type == SDL_QUIT )
        {
            //Quit the program
            quit = true;
        }
    }
}
```

As you can see, despite the fact that we're showing the messages based on key presses we don't check for any key events.

```
//Get the keystates
Uint8 *keystates = SDL_GetKeyState( NULL );
```

Instead what we do is use `SDL_GetKeyState()`.

What `SDL_GetKeyState()` does is give us the key state array. The key state array is a list of every key and whether the key is pressed or not, kind of like this:

```
SDLK_UP unpressed
SDLK_DOWN pressed
SDLK_RIGHT unpressed
SDLK_LEFT unpressed
SDLK_INSERT pressed
SDLK_HOME unpressed
SDLK_END pressed
SDLK_PAGEUP unpressed
SDLK_PAGEDOWN unpressed
SDLK_F1 unpressed
SDLK_F2 unpressed
SDLK_F3 unpressed
SDLK_F4 unpressed
SDLK_F5 unpressed
```

Now we can tell which key is down.

Just for information's sake, the argument we give `SDL_GetKeyState()` gets the number of keys available. Since we don't care about how many keys there are, we just set it to `NULL`.

```
//If up is pressed
if( keystates[ SDLK_UP ] )
{
    apply_surface( ( SCREEN_WIDTH - up->w ) / 2, ( SCREEN_HEIGHT / 2 - up->h ) / 2, up, screen );
}
```

```
//If down is pressed
if( keystates[ SDLK_DOWN ] )
{
    apply_surface( ( SCREEN_WIDTH - down->w ) / 2, ( SCREEN_HEIGHT / 2 - down->h ) / 2 + ( SCREEN_HEIGHT / 2 ), down, screen );
}

//If left is pressed
if( keystates[ SDLK_LEFT ] )
{
    apply_surface( ( SCREEN_WIDTH / 2 - left->w ) / 2, ( SCREEN_HEIGHT - left->h ) / 2, left, screen );
}

//If right is pressed
if( keystates[ SDLK_RIGHT ] )
{
    apply_surface( ( SCREEN_WIDTH / 2 - right->w ) / 2 + ( SCREEN_WIDTH / 2 ), ( SCREEN_HEIGHT - right->h ) / 2, right, screen );
}

//Update the screen
if( SDL_Flip( screen ) == -1 )
{
    return 1;
}
```

Here's a basic if up is pressed show the up message, if down is pressed show the down message, etc.

Had this program been done with events, the code would be a good size longer.

SDL_GetKeyState() and other state functions like SDL_GetModState(), SDL_GetMouseState(), SDL_JoystickGetAxis() and others can be incredibly useful. Learn more about them in the SDL API reference, which you should have handy on your computer.

Download the media and source code for this tutorial [here](#).

Quantum Research Group Leader in capacitive touch sensors. Buttons, wheels, sliders & screens.	Custom Protein Arrays 95% Pure, Active, Full-Length Recombinant Proteins & Antibodies
--	---

[News](#)
[FAQs](#)
[Games](#)
[Tutorials](#)
[Articles](#)
[Contact](#)
[Donations](#)

Copyright Lazy Foo' Productions 2004-2008