

Lazy Foo' Productions - Event Driven Programming

X out the
window!

Last Updated 1/04/14

Besides just putting images on the screen, games require that you handle input from the user. You can do that with SDL using the event handling system.

```
//Main loop flag bool quit = false; //Event handler SDL_Event e;
```

In our code after SDL is initialized and the media is loaded (as mentioned in the [previous tutorial](#)), we declare a quit flag that keeps track of whether the user has quit or not. Since we just started the application at this point, it is obviously initialized to false.

We also declare an [SDL_Event](#) union. A SDL event is some thing like a [key press](#), [mouse motion](#), [joy button press](#), etc. In this application we're going to look for quit events to end the application.

```
//While application is running while( !quit ) {
```

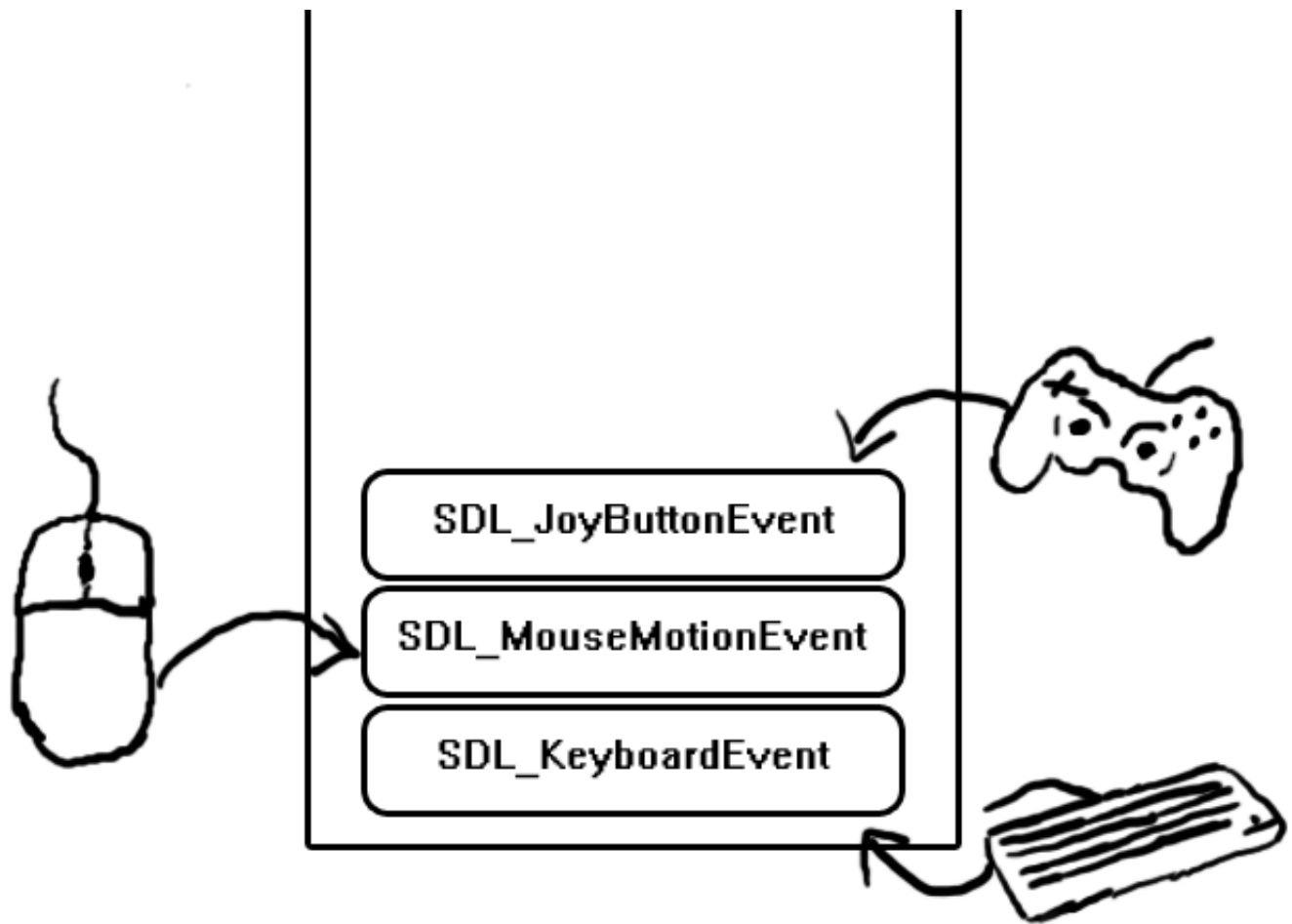
In the previous tutorials, we had the program wait for a few seconds before closing. In this application we're having the application wait until the user quits before closing.

So we'll have the application loop while the user has not quit. This loop that keeps running while the application is active is called the main loop, which is sometimes called the game loop. It is the core of any game application.

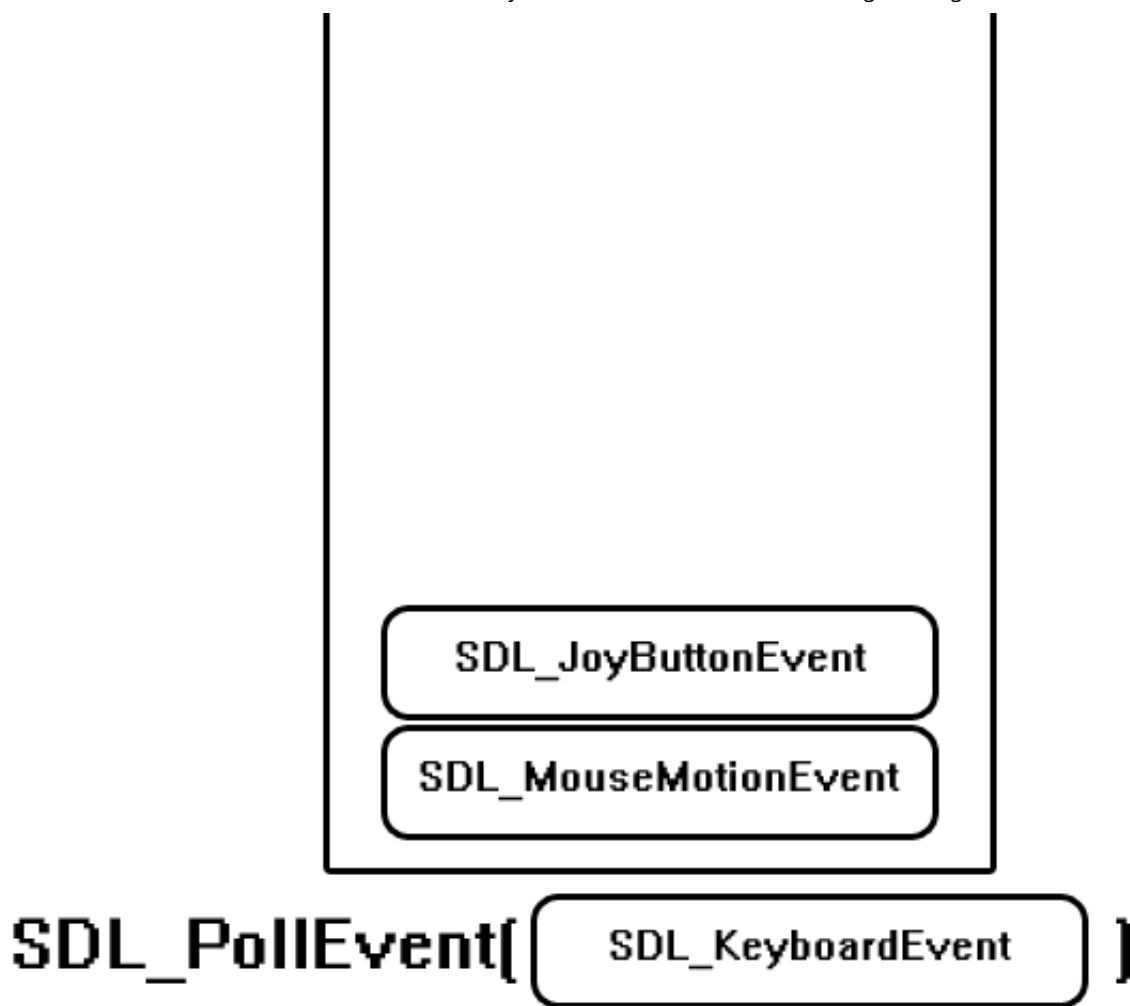
```
//Handle events on queue while( SDL_PollEvent( &e ) != 0 ) { //User requests quit if( e.type ==  
SDL_QUIT ) { quit = true; } }
```

At the top of our main loop we have our event loop. What this does is keep processing the event queue until it is empty.

When you press a key, move the mouse, or touch a touch screen you put events onto the event queue.



The event queue will then store them in the order the events occurred waiting for you to process them. When you want to find out what events occurred so you can process them, you poll the event queue to get the most recent event by calling [SDL_PollEvent](#). What `SDL_PollEvent` does is take the most recent event from the event queue and puts the data from the event into the `SDL_Event` we passed into the function.



`SDL_PollEvent` will keep taking events off the queue until it is empty. When the queue is empty, `SDL_PollEvent` will return 0. So what this piece of code does is keep polling events off the event queue until it's empty. If an event from the event queue is an `SDL_QUIT` event (which is the event when the user Xs out the window), we set the quit flag to true so we can exit the application.

```
//Apply the image SDL_BlitSurface( gXOut, NULL, gScreenSurface, NULL ); //Update the surface
SDL_UpdateWindowSurface( gWindow ); }
```

After we're done processing the events for our frame, we draw to the screen and update it (as discussed in the [previous tutorial](#)). If the quit flag was set to true, the application will exit at the end of the loop. If it is still false it will keep going until the user Xs out the window.

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