# Event-driven Programming

### Definition

Event-driven programming is a standard in which the operation and general functioning of the program is determined by direct interaction by outside elements e.g. user input, output from other programs/threads and even third-party interactions. Event-driven programming is the primary standard used in graphical user interfaces (GUIs) and other applications (e.g. JavaScript web applications) that are focused on activating specific actions in response to inputs.

### Principles

In an event-driven application, there is generally a main loop that listens for events, and then triggers a callback function when one of those events is detected. In embedded systems the same may be achieved using hardware interrupts instead of a constantly running main loop. Event-driven programs can be written in any programming language, although the task is easier in languages that provide high-level abstractions, such as closures.

### Characteristics

The defining characteristic of an event-driven program is ultimately how the program handles user input.

An event-driven program spends the majority of its’ time idle

A procedural program differs from an event-driven program in it performs the intended actions and then closes. There may be conditional statements, or events that require the program to pause while waiting for user input; but ultimately execution of the program can be followed from beginning to end. This type of program will typically display a menu and then wait until the user selects an option, then the program will display another menu and wait again until the user selects an option etc.

A Win32 program starts up, initializes a few variables, creates a window or two and then essentially stops. It spends the rest of its time in what's called a "message pump." Basically, it sits there doing absolutely nothing until something interesting happens. It creates a window, but waits until the OS tells it its time to draw it. When the OS tells it to move the window, it mvoes the window. When the OS tells it to draw the window, it draws the window. When the OS tells it a user clicked on an item, it responds. It's forever responding to events rather than making things happen.

### Features of event driven programming

Fortunately for today’s computer users, operating system software is available that allows them to use a mouse or other pointing device to select pictures, or icons, on the screen. This type of environment is a graphical user interface, or GUI. Performing an operation on an icon (for example, clicking or dragging it) causes an event—an occurrence that generates a message sent to an object.

GUI programs are called event-driven or event-based because actions occur in response to user-initiated events such as clicking a mouse button.

For the programmer, event-driven programs require unique considerations. With event-driven programs, the user might initiate any number of events in any order.

A component from which an event is generated is the source of the event. An object that is “interested in” an event you want it to respond to is a listener.

Although event-based programming is relatively new, the instructions that programmers write to correspond to events are still simply sequences, selections, and loops.