

Lecture 24

Chapter 15 Event-driven Programming

15.3 The **Event** Class

15.4 Programming Using Events

15.3 The Event Class

Event Handlers (`EventHandler` class in our graphics module) handle each individual type of event when triggered.

Recall that the signature of the `handler` method had two parameters: `self`, `event`

```
def handle(self, event)
```

`event` parameter is an instance of `Event` class.

Let's learn more about the Event class.

15.3 The Event Class

Methods of Event Class:

`getTrigger(self)`

returns a reference to the object that triggered the event (a canvas or a drawable object)

`getDescription(self)`

returns a text description of the event (for example: 'mouse click', 'mouse release', 'mouse drag', 'keyboard', 'timer')

`getMouseLocation(self)`

returns a Point designating the location of the mouse at the time of the event

`getOldMouseLocation(self)`

returns a Point designating the location of the mouse at the start of a mouse drag.

`getKey(self)`

returns a string designating the key pressed for a keyboard event

15.3 The Event Class

Mouse Events:

There are several distinct kinds of mouse events: the mouse can be single clicked, released, or dragged while the button is being held down.

We will use the following methods:

`getDescription(self)`

returns a text description of the event (for example: 'mouse click', 'mouse release', 'mouse drag', 'keyboard', 'timer')

`getMouseLocation(self)`

returns a Point designating the location of the mouse at the time of the event

15.3 The Event Class

Mouse Events - Example 1:

Let's write a program that will notify us which event happened (i.e. 'mouse click', 'mouse release', or 'mouse drag')

Sketch of the program:

```
def main():
    paper=Canvas(700,600,'light yellow','Title')
    text1=Text('Event:',12,Point(140,550))
    text1.setFontColor('dark blue')
    paper.add(text1)

    text2=Text('',12,Point(300,550))
    paper.add(text2)

    # create the handler
    mouseEvent=IdentifyMouseEventHandler(text2)

    # activate the handler
    paper.addHandler(mouseEvent)
```

See program [mouse_events.py](#)

15.3 The Event Class

Mouse Events - Example 2:

Let's slightly modify our program from Examp1 1 - let's display the coordinates too.

The change will happen only in the [handle](#) method/function.

See program [mouse_events2.py](#)

15.3 The Event Class

Mouse Events - Example 3:

Let's write a program for a dartboard.

Sketch of the program:

```
def main():  
    paper=Canvas(700,600,'light yellow','Dart Board')  
    # draw Dart Board  
    # adding comment for user  
    text=Text('Click on the dartboard to send a dart')  
    paper.add(text)  
    # draw Exit button  
    #throwing darts  
    darts=DartsHandler(paper) # creating darts handler  
    exitB=ExitButtonHandler(text) # Exit handler  
    paper.addHandler(darts) # activating darts handler  
    exitButton.addHandler(exitB) # Exit handler
```

See program [dartboard.py](#)

15.3 The Event Class

Keyboard Events:

When a user presses a key on the keyboard, this triggers a keyboard event upon whatever object currently has the “focus”. This type of event is reported as 'keyboard' by `getDescription()`.

If needed, the `getMouseLocation()` is supported for a keyboard event.

`getKey()` method returns the single character that was typed on the keyboard to trigger the event.

Example: program that echoes characters in the graphics window until the user clicks a mouse. If the mouse is clicked, everything is erased, and the user can type in a new sentence:
`echo-keyboard.py`

15.3 The Event Class

Monitor Class

Supports two methods:

`wait()`

when called, control of that flow will not be returned until the monitor is somehow released, presumably by some event handler.

`release()`

releases a monitor

This class can be thought of as “monitoring” some condition and alerting us one that condition is met.

Example: Let's write a program with two shapes (circle and rectangle) and when the user clicks on one of them the graphics window changes its background color.

See program [monitor-example.py](#)